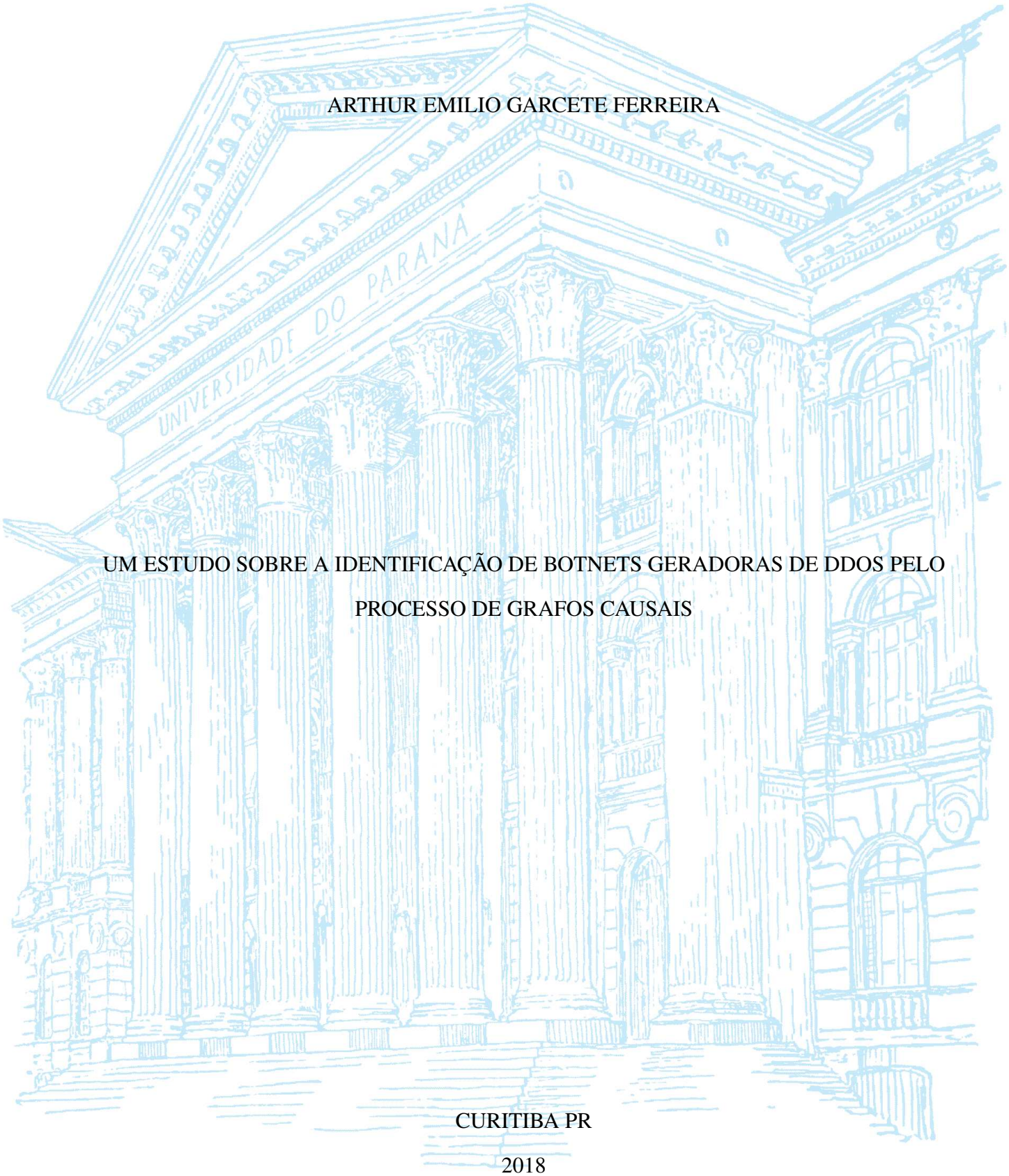
UNIVERSIDADE FERDERAL DO PARANÁ

ARTHUR EMILIO GARCETE FERREIRA



UM ESTUDO SOBRE A IDENTIFICAÇÃO DE BOTNETS GERADORAS DE DDOS PELO PROCESSO DE GRAFOS CAUSAIS

CURITIBA PR

2018

ARTHUR EMILIO GARCETE FERREIRA

UM ESTUDO SOBRE A IDENTIFICAÇÃO DE BOTNETS GERADORAS DE DDOS PELO PROCESSO DE GRAFOS CAUSAIS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Profª. Dra. Michele Nogueira Lima.

CURITIBA PR

2018

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

F383e Ferreira, Arthur Emilio Garcete
Um estudo sobre a identificação de Botnets geradoras de DDoS pelo processo de grafos causais [recurso eletrônico] / Arthur Emilio Garcete Ferreira. – Curitiba, 2018.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2018.

Orientador: Michele Nogueira Lima.

1. Processamento de sinais - métodos estatísticos. 2. Redes de computadores - controle de acesso. 3. Ataque de Negação de Serviço Distribuído. I. Universidade Federal do Paraná. II. Lima, Michele Nogueira. III. Título.

CDD: 005.87

Bibliotecária: Vanusa Maciel CRB- 9/1928

MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **ARTHUR EMILIO GARCETE FERREIRA** intitulada: **Um Estudo sobre a Identificação de Botnets Geradoras de DDoS pelo Processo de Grafos Causais**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 22 de Agosto de 2018.

MICHELE NOGUEIRA LIMA
Presidente da Banca Examinadora (UFPR)

RENATO JOSÉ DA SILVA CARMO
Avaliador Interno (UFPR)

JOSÉ AUGUSTO MIRANDA NACIF
Avaliador Externo (UFV)

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

Aos meus pais, minha fundação...

AGRADECIMENTOS

Esta é sem dúvidas a página mais emocionante de escrever. Devido a tudo o que ocorreu no período deste curso de mestrado, escolhi separar os agradecimentos em uma ordem que não necessariamente expressa a importância deles. Em primeiro lugar quero agradecer à meus pais minha mãe Maria Valentina Garcete Amarilla e meu pai José Augusto dos Santos Ferreira, pelo apoio emocional e financeiro no período em que se fez necessário, este título também é dos Sr's. Em segundo lugar a meus irmãos Frederico Emilio Garcete Ferreira, Mirella Valentina Garcete Ferreira, Edléia Alves Ferreira, Paulo Henrique Alves Ferreira, Mario Emilio Alves Ferreira, Paulo Emilio Alves Ferreira, Katherine Esther Fetsch Ferreira e também à Janice Alves Ferreira. Vocês não fazem idéia do quanto o apoio direto ou indireto foi fundamental nessa jornada. Agradeço também por meus novos irmãos, adotados nesta jornada, Paulo Lenz Junior, Gustavo Henrique de Carvalho Oliveira, Carlos Pedrozo, Danilo Rodrigo Possati, Benevid Felix da Silva, Mateus Pelloso, Rafael Araújo, Nelson Prates Júnior, Lígia Fernandes Borges, Fernando Nakayama, Renato Melo, Santiago Viertel. Não tenho palavras para descrever o quão importante vocês foram e ainda são na minha vida, obrigado por tudo, principalmente pelo apoio emocional em um período específico da minha vida que não vem ao caso agora (haha). Os "rolês" que ficaram para a história e os vídeos que é melhor nem comentar (xD). Sério, vocês não fazem idéia do amor que sinto por todos vocês. Não posso deixar de agradecer minha orientadora Michele Nogueira Lima e os professores Aldri Luiz dos Santos, Renato Carmo, Paulo Justininano Ribeiro Junior, Luiz Eduardo de Oliveira e José Nacif. Cada um de vocês tiveram um papel fundamental no meu amadurecimento profissional e acadêmico. Por isso deixo registrado aqui meu eterno agradecimento pela paciência, disponibilidade, prestatividade e comprometimento nos assuntos que envolveram minha formação no período em que aqui estive. Alguns dos sr's podem até mesmo não lembrar, mas eu lembro de cada contribuição que tiveram na minha formação. Desde um "É, eu não utilizaria este termo neste caso, acredito que (termo) se enquadra melhor por estes motivos (...)" até um "Então, por mim está tranquilo, para o seu bem e sua formação é que não aconselho apressar". Ou mesmo a frase "A matemática é uma linguagem criada para expressar o mundo". Que sem dúvidas ampliaram meus horizontes sobre os assuntos tratados durante o curso. Por isso, muito obrigado! Noedi Barbosa e Rui Barbosa, meus "pai e mãe" adotivos, não tenho palavras para expressar a gratidão do apoio e companhia de vocês durante minha estadia em sua casa. Aquele chá de laranja com mel que "mata a gripe" com bomba atômica, os bolos, doces, churrascos, doses (risos), conversas e incentivos nos tempos difíceis. Enfim, muito obrigado!

RESUMO

A Internet disponibiliza serviços de rede tendo sido seu crescimento impulsionado pelo emprego de dispositivos com acesso a essa grande rede de computadores, como *tablets*, *smartphones* e outros. O principal serviço de rede oferecido pela Internet é a conectividade, isto é, ela permite que dispositivos com características distintas troquem dados. A troca de dados segue protocolos por aplicações finais, como de mensagem instantânea, correio eletrônico e navegadores. A falha na conectividade gerada por ataques a esses protocolos causa um impacto negativo em diferentes setores da sociedade, como financeiro e até mesmo político. Dentre os diversos ataques existentes na Internet, os ataques distribuídos de negação de serviço (do inglês, *Distributed Denial of Service* – DDoS) têm se mostrado um dos mais desafiadores, tendo como objetivo comprometer o acesso a serviços oferecidos na Internet. Nesse ataque, diversos dispositivos são infectados (*bots*) e coordenados em rede (*botnet*) para executar ações maliciosas, tais como gerar uma sobrecarga contra as redes e servidores esgotando seus recursos e indisponibilizado seus serviços. Vários trabalhos na literatura abordam a detecção de *botnets*. A maioria deles detecta as *botnets* através do uso de aprendizado de máquina, onde classificadores treinados com o comportamento considerado normal da rede detectam alguma anomalia. O treinamento desses classificadores assume um estado pré-determinado da rede defendida e seus dispositivos. Isto torna essas abordagens inviáveis dependendo do tamanho e do comportamento geral dos dispositivos da rede, como a entrada e a saída constante de dispositivos. Outras abordagens utilizam o cálculo de entropia ou mesmo o cálculo do coeficiente de correlação de Pearson para diferenciar um ataque DDoS de um aumento repentino no tráfego da rede. Este trabalho estuda a eficácia do uso do processo causal em grafos (do inglês, *Causal Graph Process* - CGP) na detecção e identificação de *botnets*. O CGP é fundado em autorregressão aplicada sobre séries temporais, contendo observações de dados gerados por dispositivos durante um período para estimar a estrutura de relacionamento entre eles. As estruturas de relacionamento entre os dispositivos estimadas pelo CGP auxiliam na identificação de uma *botnet* sem a necessidade de conhecimento prévio da estrutura da rede. O CGP é aplicado em quatro bases de dados diferentes referentes a cenários de ataques DDoS distintos. Os resultados mostram que o CGP identifica as *botnets* em um curto período após o início do ataque, o que possibilita a identificação e a tomada de contramedidas de forma online, isto é, de forma que o CGP possa ser aplicado continuamente na rede para a identificação dos *bots*. Os trabalhos futuros terão foco na otimização do algoritmo para a identificação de *botnets*.

Palavras-chave: Processamento de Sinais, Segurança de Redes, Botnet, Ataque de Negação de Serviço Distribuído

ABSTRACT

The Internet provides network services and its growth has been supported by the increasing popularization of mobile devices, such as tablets, smartphones, and others, having access to the wide computer network. The main network service provided by the Internet is connectivity. This allows heterogeneous devices to exchange data through protocols. The protocols allow end-to-end applications, such as instant messaging, e-mail and browsing, to exchange data. Thus, connectivity failures caused by an attack against these protocols generate great impact in many society sectors, such as financial and political. Among the several existent threats on the Internet, one of the most challenging is the Distributed Denial of Service (DDoS) attack. Its goal is to compromise access to services provided on the Internet. In this attack, the attacker uses several infected hosts (bots) coordinated in a network (botnet) to perform malicious activities, such as coordinating an overload against networks and servers, running out its resources and making it unavailable. There are several approaches in the literature to detect botnets. Most of the approaches detect botnet activity by Machine Learning methods, in which a classifier is trained with the normal network behavior to detect anomalies. The training of these classifiers lies in the main assumption that the network behavior is always the same. Thus, these approaches are not feasible to defend a network depending on its size and its behavior, such as the constant network devices input and output. Other approaches use the entropy of the network behavior or even the Pearson correlation coefficient to discriminate a DDoS attack behavior from a flash-crowd event. This work investigates the use of Causal Graph Process in the detection and identification of botnets. CGP is based on an autoregression over temporal series containing observations generated by hosts to estimate the relationship structure between hosts. The relationship between hosts, estimated by the CGP, assists in botnet identification process. CGP does not require prior knowledge about the network. It is applied on four different datasets containing distinct DDoS attack scenarios. Results show that the CGP is able to assist in the botnet detection shortly after the attack starts, allowing online countermeasures to take part in the defense system. Future works will take place on enhancing CGP to identify botnets.

Keywords: Signal Processing on Graphs, Computer Networks, Network Security, Botnet, Distributed Denial of Service

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

CGP *Causal Graph Process*
Processo Causal Em Grafos

WBAN *Wireless Body Area Networks*
Redes Sem Fio Corporais

IoT *Internet of Things*
Internet Das Coisas

DoS *Denial of Service*
Ataque De Negação De Serviço

DDoS *Distributed Denial of Service*
Ataque Distribuído De Negação De Serviço

C&C *Command & Control*
Comando E Controle

IP *Internet Protocol*
Protocolo De Internet

TCP *Transimssion Control Protocol*
Protocolo De Controle De Transmissão

ICMP *Internet Control Message Protocol*
Protocolo De Controle De Mensagens Da Internet

HTTP *HyperText Transfer Protocol*
Protocolo De Transferência De HiperTexto

IRC *Internet Relay Chat*
Relé De Bate-papo De Internet

API *Application Interface*
Interfaces De Aplicação

IDS *Intrusion Detection System*
Sistemas De Detecção De Intrusão

P2P *Peer-to-Peer*
Par-a-par

ACF *Auto Correlation Function*
Função De Autocorrelação

RLM Regressão Linear Múltipla

SQT Soma Dos Quadrados Totais

SQR Soma Dos Quadrados Dos Resíduos

SQE Soma Dos Quadrados Explicados

GPSR *Gradient Projection for Sparse vector Reconstruction*
Projeção Gradiente Para Reconstrução De Vetores Dispersos

SVM *Support Vector Machine*
Máquina De Vetores De Suporte

SVM *Support Vector Machine*
Máquina De Vetores De Suporte

DT *Decision Tree*
Árvore De Decisão

HEMST *Hierarchical Euclidean Minimum Spanning Tree*
Árvore Hierárquica Euclidiana Geradora Mínima

GTS *Generate Time Series*
Gerar Série Temporal

CSV *Comma Separated Values*
Valores Separados Por Vírgulas

CAIDA *Center for Applied Internet Data Analysis*
Centro Para Análise Aplicada De Dados Da Internet

CTU *Czech Tecnical University*
Universidade Tecnológica Tcheca

SLS Secure Linux Solutions

RTT *Round Trip Time*
Tempo De Ida E Volta De Um Pacote

LISTA DE SÍMBOLOS

X	conjunto da quantidade de observações da variável y
Y	conjunto de observações da variável y
y	variável observada
α	onde o eixo Y intercepta o eixo X
β	variação de x_t em relação a y_t
t	amostra ou observação em determinado intervalo
e	porção da função não explicada por α e β
p	ordem da regressão
n	quantidade de variáveis
$\bar{y}$	média dos valores observados da variável y
$\hat{y}$	valor estimado pelo modelo quando x_t
r^2	coeficiente de determinação da regressão
$\mathbf{A}$	matriz de adjacência
$\hat{\mathbf{A}}$	estimativa da matriz de adjacência
$\hat{\mathbf{R}}$	matriz de convolução entre as autorregressões de X

1 INTRODUÇÃO

Nos últimos anos, a Internet tem se tornado um dos principais meios de comunicação e de informação no mundo [Diego et al., 2018, Zhou e Ye, 2017]. Em 1997, [Cronin, 1997] estimou a importância que a Internet teria nos dias atuais. Recentemente, [Lupien et al., 2017] confirmaram esta previsão realizando um estudo sobre o impacto socioeconômico da Internet na atualidade. Este impacto deve-se em parte à popularização de dispositivos que usam seus protocolos de comunicação para transmitir e receber dados. A Internet é em sua essência uma rede de redes conectadas ao redor do mundo, o que faz dela uma rede com alcance mundial. Essa rede foi inicialmente composta apenas por computadores. Porém, devido à expansão tecnológica dos últimos anos, outros dispositivos computacionais vêm sendo conectados à rede, tais como *smartphones*, *smartwatches* e câmeras de vigilância. Devido à sua abrangência geográfica, as empresas têm disponibilizado serviços através da Internet. O Internet Banking, os portais de notícias e o ensino à distância são exemplos de aplicações que usam os serviços de Internet para trafegar dados. Em alguns casos, como nas Redes Sem Fio Corporais (do inglês, *Wireless Body Area Networks* - WBAN), a disponibilidade dos serviços de rede é um problema crítico pois falhas podem levar a diagnósticos errados e até mesmo colocar em risco a vida de pessoas [Liu e Kwak, 2010]. De acordo com o estudo realizado por [Sachdeva et al., 2008], a indisponibilidade de serviços na rede causa em média um prejuízo financeiro de aproximadamente US$ 60.000.000,00 por ano. Isso torna a segurança de redes um assunto cada vez mais importante, uma vez que o uso da Internet para aplicações críticas tem crescido tal como a Internet das Coisas (do inglês, *Internet of Things* - IoT) [Arun e Selvakumar, 2009].

Três pilares compõem os estudos voltados para segurança de redes, são eles a integridade, a confidencialidade e a disponibilidade. A integridade tem por objetivo garantir que os dados trafegados pela rede sejam autênticos, isto é, não sofreram alterações no percurso de sua origem até seu destino. A confidencialidade visa garantir que os dados só possam ser inteligíveis pelos dispositivos envolvidos diretamente na comunicação, isto é, somente os remetentes e os destinatários legítimos podem ter acesso ao conteúdo dos dados trafegados de forma clara. A disponibilidade versa na garantia do funcionamento adequado dos meios de comunicação da rede, isto é, na garantia de que os dados tenham um ambiente estável para trafegar sem que haja perdas ou atrasos. Qualquer tipo de distúrbio intencional em um desses três pilares é considerado um ataque e deve ser tratado.

Na Internet, um dos tipos de ataque mais comum é o Ataque de Negação de Serviço (do inglês, *Denial of Service* - DoS) [Mansfield-Devine, 2016], que visa comprometer o funcionamento adequado dos serviços e sua disponibilidade. Uma variação do ataque DoS é o Ataque Distribuído de Negação de Serviço (do inglês, *Distributed Denial of Service* - DDoS). Este compromete a disponibilidade de serviços em uma rede ou servidor alvo através de um grande conjunto de dispositivos distribuídos e coordenados em rede (*Botnet*). Logo, um desafio na área de Segurança de Redes é assegurar o funcionamento adequado das redes locais e, por extensão, da Internet.

1.1 PROBLEMA

O crescimento exponencial do número de dispositivos com acesso à Internet é um indicador de que as *botnets* geradoras de DDoS merecem atenção [Perakovic et al., 2015]. Isto porque a vulnerabilidade originada das limitações técnicas dos dispositivos da IoT tornam

dispositivos dessa categoria alvos de infecção, podendo torná-los *bots* e expandindo a abrangência das *botnets* [Angrishi, 2017]. Os atacantes exploram essas vulnerabilidades a fim de aumentar o tamanho da sua *botnet* para realizar ataques DDoS [Bertino e Islam, 2017]. Dadas as diversas aplicações da IoT, como uma aplicação médica, vigilância digital e sensoriamento térmico, a criação de métodos eficazes que sejam capazes de defender de forma online e reativa as redes nas quais esses dispositivos atuam se faz cada vez mais necessária. A defesa online é caracterizada pela não necessidade de isolamento do tráfego de rede para posterior anlálise e tomadas de medidas defensivas. A defesa reativa é caractarizada pela reação do mecanismo de defesa ao identificar um ataque a rede. Isto porque essas aplicações requerem uma rede com alta disponibilidade, isto é, uma rede que esteja apta a transportar os dados de maneira eficaz sem interferências externas.

As *botnets* são utilizadas na execução de ataques coordenados na Internet. Os recursos atingidos por esses tipos de ataque são a largura de banda, memória e capacidade de processamento (servidor). Estes recursos estão diretamente relacionados à disponibilidade da rede. Assim, a detecção e a identificação de *botnets* são importantes devido ao dano potencial causado por seus ataques. Para isto, são necessários métodos que sejam capazes de detectar e identificar *botnets* para evitar o sucesso de um ataque gerado por ela, isto é, para que as contramedidas sejam acionadas antes da exaustão dos recursos da vítima, visando manter a disponibilidade da rede e o funcionamento dos serviços oferecidos por empresas que usam a Internet para prover seus produtos e serviços.

A detecção de *botnets* tem por objetivo verificar a existência de atividade de *botnet* na rede, mas não necessariamente identifica quais dispositivos da rede são os *bots*. A identificação da *botnet* tem como objetivo apontar os *bots*, isto é, discriminar quais dispositivos estão atacando a rede. A detecção e a identificação são realizadas com base no comportamento desses dispositivos durante um ataque. A detecção é o primeiro passo na identificação de uma *botnet*, pois permite que o administrador de rede saiba que a indisponibilidade da rede está sendo causada por um ataque. Assim, a identificação da *botnet* é um segundo passo, onde técnicas de classificação e mineração de dados são utilizadas a fim de distinguir os dispositivos atacantes da rede.

1.2 MOTIVAÇÃO

As técnicas de detecção de *botnets* atuais necessitam de um treinamento prévio do comportamento considerado normal da rede [Bhuyan et al., 2015]. Contudo, essas técnicas exigem um treinamento contínuo do comportamento considerado normal da rede para que sejam eficazes, isto é, o processo de treinamento dessas técnicas, que é computacionalmente custoso, pode por vezes tornar essas abordagens inviáveis dependendo da variação do comportamento ou da estrutura da rede. Isto considerando que esses métodos apenas detectam a atividade de uma *botnet* na rede e não necessariamente as identifica. Desta forma, se faz necessário um método capaz de detectar e identificar as *botnets* de forma contínua (ou online) sem a necessidade de um estágio de aprendizado e treinamento a respeito do comportamento normal da rede. Na literatura, abordagens recentes tratam desse assunto, onde em sua maioria utilizam técnicas de aprendizado de máquina para detectar e mitigar os efeitos da *botnet* [Feily et al., 2009].

Os estudos têm direcionado o desenvolvimento de sistemas capazes de detectar as atividades da *botnet* na rede de maneira eficaz utilizando vários métodos, como a busca por padrões de comunicação de Comando e Controle (do inglês, *Command & Control* - C&C), correlação entre pares de endereços baseado nos fluxos de dados e pela estimativa da quantidade de *bots* (dispositivos infectados) durante um ataque através do valor da entropia da rede. Em termos gerais, a entropia é um valor calculado para expor a desordem de uma rede com base em

suas características, quanto maior a variação deste valor, maior é a desordem da rede e portanto, maiores as chances desta rede apresentar anomalias [Kalaivani e Vijaya, 2016].

Como já mencionado, essas abordagens podem ser inviáveis como soluções online pois o estágio de treinamento é custoso e por vezes não é rápido o suficiente a ponto de assegurar a detecção e a identificação das *botnets*. O custo computacional considerado ideal para uma solução online é denotado na terminologia big-O como $O(1)$, ou seja, dada uma entrada a saída é imediata [Farines et al., 2000]. Atualmente não há implementação de algoritmo relacionado às técnicas de segurança de redes e detecção e identificação de *botnets* que apresente complexidade $O(1)$, e portanto podem não atender à demanda de segurança dessas novas aplicações. Estas técnicas, por considerarem em sua maioria um estado imutável da rede, podem não ser eficazes se abordadas como método de identificação contínuo de *botnets*. Logo, sua eficácia é limitada ao aprendizado já realizado e o custo do treinamento, se aplicado de forma contínua (ou online) pode ser inviável dependendo da variação natural do comportamento da rede.

Este trabalho é motivado pela necessidade de um método capaz de detectar e identificar *bots* e *botnets* de forma online e reativa e que seja indiferente a mudanças na estrutura da rede. Isto é, que não necessite de um estágio de treinamento com o comportamento normal da rede para que alguma anomalia seja detectada e que, posteriormente, a *botnet* seja identificada. Esta identificação deve ocorrer preferencialmente antes que o ataque tenha efeito a fim de que as contramedidas, como isolamento de tráfego, filtragem de pacotes e limitação de requisições de determinada origem sejam tomadas antes que a disponibilidade da rede seja comprometida, evitando possíveis danos, como a interrupção dos serviços. Diferente dos outros trabalhos na literatura, que focam principalmente na detecção da atividade da *botnet* na rede, este foca no estudo de um método que também identifica os *bots*.

Este trabalho analisa a eficácia do CGP na identificação de *botnets*. O CGP, quando adaptado para a aplicação em redes, deve ser indiferente a mudanças na estrutura da rede a fim de ser aplicado em qualquer tipo de rede além de permitir que a detecção seja realizada de forma online e reativa, e que contribua para a manutenção da disponibilidade da rede. O CGP utiliza um modelo estendido de processamento de sinais em grafos que estima as interrelações entre os sinais emitidos por dispositivos heterogêneos e representa as interrelações através de um grafo direcionado ponderado. A autocorrelação é calculada através de uma série temporal de entrada que contém observações de dispositivos durante um período. O período é composto por uma série de intervalos menores contendo dados observados desses dispositivos, como a soma do tamanho de pacotes ou a quantidade de pacotes. Essas observações dentro de cada intervalo são chamadas de amostras, que por sua vez são parâmetros configuráveis, fazendo o método se enquadrar nos requisitos de solução online e reativa. Isto porque os ajustes destes parâmetros não requerem a interrupção da rede para terem efeito, permitindo a execução do CGP de forma contínua.

1.3 OBJETIVOS E CONTRIBUIÇÕES

Este trabalho tem como objetivo contribuir com os estudos relacionados à segurança de redes através de um estudo sobre o uso da técnica de processo causal em grafos (do Inglês, *Causal Graph Process* - CGP) como método capaz de identificar *botnets* geradoras de ataques DDoS de forma online e sem a necessidade de uma fase de treinamento. Este trabalho também apresenta uma adaptação do método CGP a fim de torná-lo mais adequado ao contexto de redes de computadores. A avaliação do CGP adaptado resultou na ratificação da eficácia deste método na detecção e na identificação de *botnets*. Logo, a principal contribuição deste trabalho é a

adaptação e a avaliação do método CGP como uma ferramenta de detecção e identificação de *botnets* em redes de computadores.

1.4 ESTRUTURA DO ESTUDO

O restante deste manuscrito está organizado da seguinte forma. O Capítulo 2 apresenta os trabalhos relacionados e os fundamentos. O Capítulo 3 apresenta o modelo CGP avaliado na identificação de *botnets* geradoras de ataques DDoS. O Capítulo 4 apresenta a metodologia utilizada para avaliar o CGP, onde são expostas as adaptações realizadas para utilização em redes e onde também é realzada uma análise das bases de dados onde o CGP adaptado foi aplicado; os resultados e a discussão dos mesmos. Por fim, o Capítulo 5 apresenta as conclusões, elucidando os principais pontos abordados, pontos em aberto e trabalhos futuros.

2 FUNDAMENTOS E TRABALHOS RELACIONADOS

Este capítulo apresenta os conceitos envolvidos neste trabalho associados à descrição do problema tratado e ao método investigado. A ênfase é dada ao conceito de ataque volumétrico distribuído de negação de serviço (DDoS) e conceitos de regressão por serem o foco deste trabalho. Antes disso, é apresentada a revisão bibliográfica. Este capítulo está dividido como segue. A Seção 2.1 conceitua um ataque DDoS e suas principais características além de fundamentar os componentes envolvidos em um ataque. A Seção 2.2 descreve as características de uma série temporal, contextualizando-a com o problema tratado e o método avaliado. A Seção 2.3 aborda os conceitos de correlação e causalidade aplicados na detecção de *botnets*, os fundamentos de regressão que embasam o Processo de Grafo Causal (*Casual Graph Process* – CGP). A Seção 2.4 apresenta um levantamento bibliográfico onde são abordados os trabalhos relacionados e as principais técnicas identificadas para a detecção e identificação de *botnets*. Por fim, a Seção 2.5 resume este capítulo.

2.1 AS REDES DE COMPUTADORES E A NEGAÇÃO DE SERVIÇO

O ataque distribuído de negação de serviço (DDoS) é um dos diversos tipos de ataque que visa comprometer o funcionamento das redes e servidores alvo. Existem variações deste ataque onde um único dispositivo infectado dissemina um código malicioso (*malware*) que executa rotinas para bloquear portas do dispositivo infectado. Em alguns casos um só dispositivo se passa por um roteador na rede, descartando, redirecionando ou capturando todo o tráfego de determinada rede local, podendo assim ter acesso a informação privilegiada ou mesmo modificar as informações. No caso do ataque DDoS diversos dispositivos infectados são utilizados para inundar uma rede local ou um servidor com tráfego malicioso, consumindo seus recursos e tornando-os indisponíveis para os usuários legítimos.

Esse tipo de ataque envolve diferentes entidades, como o atacante, os meios de ataque, e a vítima que pode ser a rede ou um servidor que provê um serviço. O atacante que é a pessoa ou entidade que decide e executa os comandos de ataque. Os meios de ataque compreendem os dispositivos utilizados para a realização do ataque. Esta seção descreve essas entidades que compõem um cenário de ataque DDoS e seus papéis durante o ataque. Antes disso, será apresentado o conceito de rede e seus elementos, como um serviço no contexto de Internet, servidor, cliente, *firewall*, *bot*, *botnet*, e por último, o que é um ataque DoS.

De forma genérica, uma rede de computadores é uma estrutura de dispositivos que trocam dados. Essa troca de dados ocorre através de protocolos. Os protocolos estabelecem regras a serem seguidas para que os dados sejam transmitidos entre dispositivos. Dentre os diversos protocolos destacam-se o protocolo de Internet (do inglês, *Internet Protocol* - IP) e o protocolo de controle de transmissão (do inglês, *Transimssion Control Protocol* - TCP). Esses dois protocolos em conjunto estabelecem as regras de comunicação entre dispositivos na rede de forma confiável, isto é, com mecanismos que garantem a ordem, integridade e a entrega dos dados na rede. Outro protocolo bastante utilizado na rede é o protocolo de controle de mensagens da Internet (do inglês, *Internet Control Message Protocol* - ICMP), que fornece relatórios de erro sobre uma rede ao solicitante. Os protocolos da camada de aplicação como o protocolo de transferência de HiperTexto (do inglês, *HyperText Transfer Protocol* - HTTP), relé de bate-papo de Internet (do inglês, *Internet Relay Chat* - IRC), são exemplos de protocolos que usam os protocolos TCP e IP nas camadas mais baixas para trafegar dados. O HTTP é amplamente

utilizado entre navegadores, e possibilita o acesso a *websites* e interfaces de aplicação (do inglês, *Application Interface* - API) na *web*. O IRC é um protocolo de troca de mensagens, normalmente utilizado para bate-papo. Estes, porém, definem as regras de comunicação entre aplicações que utilizam protocolos de rede para trafegar seus dados.

Na Internet, o principal serviço de rede oferecido é a transferência confiável de dados [Kurose e Ross, 2010]. A transferência confiável de dados consiste na garantia de que um dado saia de seu remetente e chegue ao seu destinatário. A partir da popularização da Internet, a segurança se tornou um serviço requisitado pelos usuários dessa grande rede de computadores. A segurança consiste na realização da comunicação de forma segura, isto é, utilizando mecanismos que possibilitem que os recursos necessários para a comunicação não sejam negados, que apenas o destinatário leia e entenda mensagem, que o remetente seja realmente o autor da mensagem, que o destinatário é realmente quem diz ser e que a mensagem recebida seja idêntica a original [Kurose e Ross, 2010]. Qualquer distúrbio ou exploração indevida dos protocolos de rede é considerado uma ameaça à segurança de redes e portanto um ataque. As aplicações de troca de mensagens, videochamadas, armazenamento de arquivos em local remoto e acesso à informação são exemplos de serviços prestados e oferecidos por empresas através da Internet que dependem dos serviços de rede para seu funcionamento.

Em uma rede, cada dispositivo desempenha uma função que contribui na disponibilização e utilização dos serviços. Dentre as funções estão o servidor, o cliente, o roteador e o *firewall*. Um servidor é um dispositivo em uma rede que disponibiliza serviços ou recursos através de um ou mais protocolos [Kurose e Ross, 2010]. Um cliente é um dispositivo que consome serviços e/ou recursos disponibilizados pelos servidores através de um ou mais protocolos. Um roteador é o dispositivo responsável pelo roteamento de pacotes na rede, isto é, pelo direcionamento do tráfego na rede. Um *firewall* é um sistema de defesa de uma rede composto por sistemas de detecção de intrusão (do inglês, *Intrusion Detection System* - IDS), *Proxys* e *Honeypots*, implementados em um ou mais dispositivos na rede [Filho, 2013]. Todos os dispositivos que estão conectados em uma rede são chamados de *hosts* [Kurose e Ross, 2010].

Em alguns casos, a transmissão de dados é realizada através de *bots*. Um *bot* é um host infectado com um ou mais *malwares* similares que utilizam protocolos como HTTP ou IRC para se comunicar e executar ações instruídas por um atacante. As instruções para a realização destas ações são retransmitidas por um agente chamado *BotMaster* que encaminha estas instruções definidas pelo atacante para os *bots* remotamente. Finalmente, uma *botnet* é uma rede composta de dois ou mais *bots* utilizada para executar ações coordenadas. Estas ações neste caso são por exemplo um ataque de negação de serviço [Feily et al., 2009].

A negação de serviço é um tipo de ataque que visa comprometer intencionalmente a disponibilidade da rede a seus usuários legítimos. Este tráfego malicioso tem como objetivo consumir os recursos da rede ou servidor tornando-os indisponíveis. Existem três principais classificações de ataques de negação de serviço na Internet sendo:

Volumétrico: Consiste de múltiplos *bots* inundando uma rede local com tráfego ilegítimo visando consumir a largura de banda, comprometendo a vazão e indisponibilizando a rede. Esta é a base dos ataque DDoS.

Zero-day: Um ou mais sistemas infectados explorando falhas (*bugs*) em sistemas finais. Essas falhas causam a interrupção ou o encerramento dos processos referentes a esses sistemas tornando-os indisponíveis.

Baixa taxa (Low-rate): Tipo de ataque que explora maliciosamente as vulnerabilidades dos protocolos de rede como temporização do envio de pacotes considerando o tamanho da

janela deslizante e suas retransmissões para causar a exaustão dos recursos de rede e assim, indisponibilizá-la.

Um ataque de negação de serviço distribuído segue geralmente três fases principais:

1. Infecção: o *BotMaster* - computador com código malicioso e rotinas para espalhamento deste código malicioso controlado por um atacante - procura *hosts* vulneráveis que possibilitem a instalação de código malicioso (*malware*). Uma vez instalado o *malware* se comunica com o *BotMaster* e aguarda ordens do atacante.

2. Manutenção: os dispositivos infectados (*bots*) estão aguardando ordens do *BotMaster* e dependendo do comportamento do *malware*, tentam infectar outros dispositivos a fim de aumentar a *botnet.* Esse é um dos estágios mais críticos, porque alguns dispositivos que possuem sistemas de segurança mais elevados podem conseguir detectar e remover o malware no sistema.

3. Ataque: o *atacante* escolhe uma vítima, normalmente um servidor na Internet, e envia um comando para o BotMaster que repassa essas instruções para a *botnet* executar um ataque. O tipo mais comum de ataque é a inundação da vítima com requisições ilegítimas a fim de esgotar os recursos da rede alvo, como largura de banda e processamento. A Figura 2.1 ilustra o diagrama de um ataque DDoS. Na figura, o atacante envia um comando ao *BotMaster*. Este utiliza todos os *bots* infectados sob seu controle para iniciar o ataque contra uma rede ou servidor alvo definido pelo atacante. As mensagens enviadas pelo acatante ao *BotMaster* e do *BotMaster* para os *bots* são chamadas de mensagens de Comando & Controle (do inglês, *Command & Control* - C&C). Essas mensagens são transmitidas através de protocolos como o IRC e HTTP, que são amplamente utilizados na Internet para troca de mensagens entre aplicações de conversação, navegadores e neste caso, *bots*. A infraestrutura da botnet ilustrada na Figura 2.1 representa um modelo hierárquico da *botnet*, onde os *bots* se comunicam exclusivamente com o *BotMaster*. Existe outra topologia ilustrada na Figura 2.2 que pode ser utilizada por uma *botnet* onde todos os *bots* têm acesso e se comunicam com os outros *bots*. Este tipo de arquitetura é conhecida como par-a-par (do inglês, *Peer-to-Peer* - P2P) e permite que qualquer *bot* na *botnet* possa atuar como *BotMaster*.

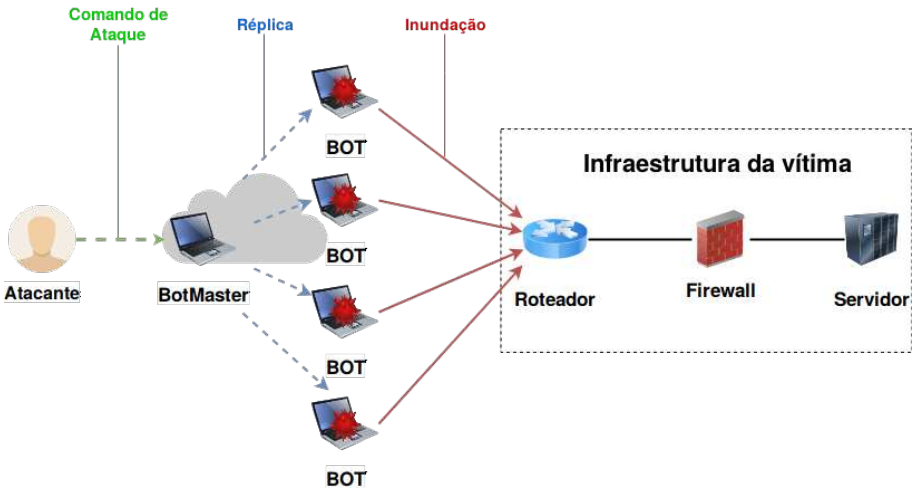


Figura 2.1: Arquitetura Hierárquica de um ataque DDoS

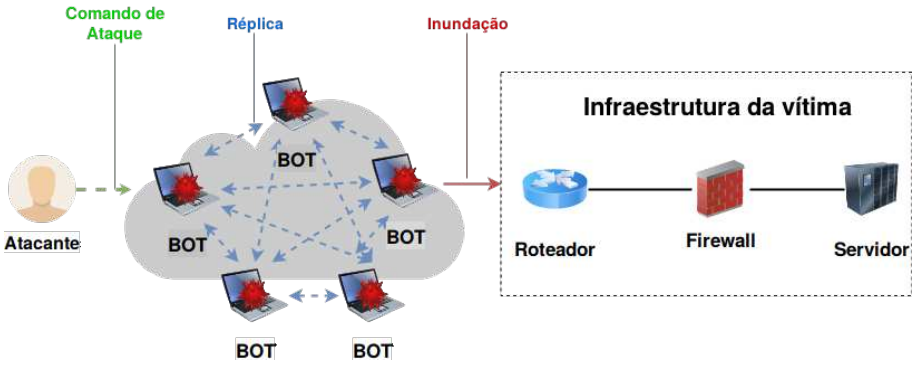


Figura 2.2: Arquitetura P2P de um ataque DDoS

Em um ataque originado por uma *botnet*, uma característica frequente é a ação do atacante ordenando o início do ataque. Por partir normalmente de um mesmo atacante, o comando de ataque replicado pelo *BotMaster* aos *bots* segue um mesmo padrão. Em um ataque DDoS o comando de ataque pode ser enviar uma quantidade específica de pacotes por determinado tempo. Isso por sua vez permite aos métodos de detecção de ataques gerados por *botnets* assumir que antes e durante um ataque, existe um padrão de comunicação detectável entre o *BotMaster* e os *bots*, ou mesmo entre os *bots* e a vítima como abordados na seção anterior. A próxima seção aborda as séries temporais, uma forma de representacao de dados que pode ser utilizada para detectar e identificar padrões de comunicação de *botnets*.

2.2 SÉRIES TEMPORAIS

Uma das formas de representação de dados utilizadas na análise de dados é a série temporal, que consiste no conjunto de observações sequenciais de uma ou mais variáveis ao longo do tempo [Montgomery et al., 2006]. A unidade de observação de uma variável em uma série temporal é chamada de amostra e contém o valor da variável em um determinado instante de tempo dentro do período observado. A quantidade de amostras determina o tamanho de uma série temporal. A periodicidade entre amostras, isto é, o intervalo em que as observações são coletadas e varia de acordo com a possibilidade de coleta dos dados dessas variáveis. Quando existem observações de apenas uma variável na série temporal, dizemos que se trata de uma série temporal univariada. Quando existe mais de uma variável, podemos dizer que se trata de uma série temporal multivariada. A Figura 2.3 ilustra duas séries temporais genéricas, uma univariada e outra multivariada, contendo 10 observações de diferentes variáveis. Os valores variam de 0 à 310. As variáveis estão discriminadas por cor, e os pontos representam as amostras observadas destas variáveis.

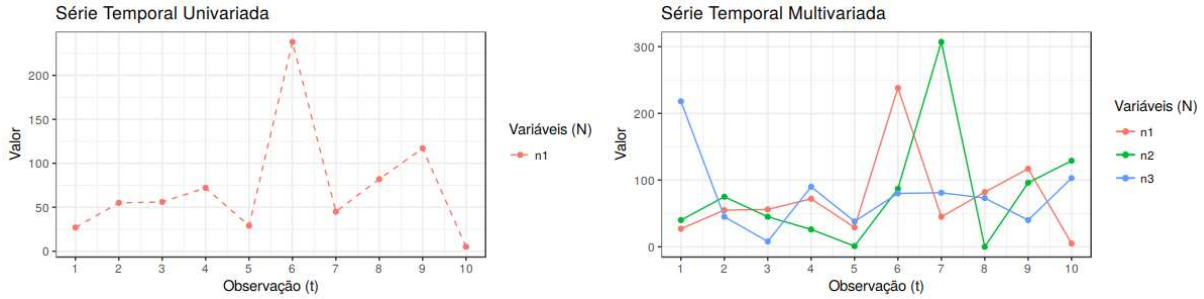


Figura 2.3: Exemplos de Séries Temporais

As séries temporais univariadas ou multivariadas são amplamente utilizadas na econometria e têm se tornado cada vez mais relevantes nas análises de dados [Montgomery et al., 2006, Masud et al., 2008]. Isso porque a análise comportamental de variáveis ao longo do tempo contribui para a análise e para a caracterização do comportamento dos *hosts* e da rede, podendo expor padrões e até mesmo predizer comportamentos desses *hosts* e da rede. Exemplos de padrões de comportamento são a coordenação de uma *botnet* durante um ataque DDoS e a relação de causa e efeito entre a vítima e os *bots* e entre os *bots*, dentre outros.

Os comportamentos de *hosts* podem ser caracterizados e até mesmo estimados através de técnicas de análise de séries temporais. Uma das técnicas é a regressão, processo que possibilita a modelagem e a caracterização do comportamento das variáveis observadas através de cálculos realizados sobre os valores observados dessas variáveis [Montgomery et al., 2006]. O processo de regressão e suas particularidades são explicados na subseção 2.3.1.

2.3 CORRELAÇÃO E CAUSALIDADE

A correlação é o nome dado à similaridade ou relação mútua entre uma ou mais variáveis [Cohen et al., 2013]. O cálculo de correlação é utilizado para estabelecer representações de relação entre variáveis. Um exemplo genérico de correlação entre duas variáveis é a distância em metros percorrida por um objeto em relação à velocidade em metros por segundo desse mesmo objeto por um tempo determinado. Na literatura, existem diversas técnicas para indicar essas relações entre uma ou mais variáveis [Cohen et al., 2013]. Essas relações são calculadas de acordo com as observações coletadas das variáveis. Em alguns casos, a correlação implica na causalidade entre essas variáveis.

A causalidade é o nome dado às relações de causa e efeito entre duas variáveis analisadas [Ltkepohl, 2007]. Isto é, o comportamento de uma variável causa determinado comportamento em outra variável. Essa relação depende do contexto e das variáveis escolhidas para serem correlacionadas. O aumento no valor de um determinado produto devido à redução da produção deste produto é um exemplo de correlação e causalidade. Já o aumento no imposto devido ao aumento na produção de um produto não é necessariamente uma relação de correlação e causalidade.

Para calcular a correlação entre variáveis é necessário que se tenha os registros das observações de ao menos uma variável. Estes registros são realizados continuamente em intervalos de tempo predeterminados. Quando isso ocorre, os registros de uma ou mais variáveis formam uma série temporal. Em uma série temporal multivariada, o cálculo de correlação ocorre entre as observações de uma variável em relação às demais variáveis observadas. Em uma série temporal univariada, o cálculo de correlação dessa variável é realizado considerando apenas suas próprias observações anteriores. Existem diversos métodos na literatura para inferir correlação entre variáveis [Cohen et al., 2013]. Neste trabalho abordamos a regressão, que é uma das técnicas que compõem o método CGP avaliado neste trabalho.

2.3.1 Regressão Linear

A regressão linear é um processo onde dada uma série temporal contendo observações sequenciais de uma variável observada, assume-se que essas observações apresentam informações sobre o seu comportamento. Ou seja, o dado observado de uma variável dentro de qualquer período é expresso através de uma função [Montgomery et al., 2006]. Essa função expressa, dadas as observações de uma variável, a relação entre ela e outras variáveis ou entre ela e seus próprios valores observados. Isto é, o quanto o valor de uma variável influencia outras variáveis

ou o quanto os seus valores anteriores influenciam seus valores atuais. A função encontrada reproduz as estimativas dos valores observados e se apresenta geralmente da seguinte forma:

$$y_t = \alpha + \beta x_t + e_t$$

Considerando as observações de uma variável representada em um gráfico bidimensional com eixos X, Y, onde $X = \{x \in \mathbb{N}\}$ indica a quantidade de observações de y, e $Y = \{y \in \mathbb{R}\}$ o valor observado da variável y, α é uma constante que indica em que parte o eixo Y intercepta o eixo X; β é outra constante que representa o quanto y_t varia em relação à variável no eixo X. x_t é a variável explicativa, que é o valor da variável do eixo X no intervalo t; e finalmente e_t é a variável que inclui todos os fatores residuais, que corresponde à porção da função não explicada por $\alpha + \beta x_t$. Ou seja, o quando x está em t, y_t varia $\alpha + \beta$ multiplicando o valor de x no intervalo t + a margem de erro e. Essa modelagem apesar de simples, é efetiva para a caracterização do comportamento das variáveis de uma série temporal [Chatfield, 2004].

Através da regressão linear é possível modelar relações entre variáveis observadas e realizar previsões com certo grau de precisão [Cohen et al., 2013]. Geralmente, quando não se usa a regressão linear, o cálculo da média dos valores observados na série temporal é utilizado para estimar uma possível nova observação [Montgomery e Runger, 2003]. Essa abordagem apesar de rápida e prática, pode não representar de forma acurada o comportamento da variável observada por ignorar fatores que a influenciam. A regressão linear é mais eficaz neste sentido por considerar estes fatores como a correlação entre variáveis dentro de um intervalo, expressados através dos coeficientes α , βx_t e e_t.

Normalmente, a regressão linear é realizada utilizando observações de duas ou mais variáveis, uma sendo a variável explicada (dependente, representada por y), e outra explicativa (independente, representada por x). Em alguns casos, têm-se apenas uma variável observada na série temporal. Quando isso ocorre, utilizam-se os valores das observações anteriores da mesma variável para a regressão. Com isso, a regressão passa a ser chamada de **autoregressão** e a função toma a seguinte forma:

$$y_t = \alpha + \beta y_{t-p} + e_t$$

Onde p representa a ordem da regressão, que corresponde à quantidade de observações anteriores de y_t utilizadas na regressão. Nestes casos, a autoregressão é condicionada a ordem que apresente autocorrelação dos valores observados na série temporal. Esta ordem (ou lag) pode ser encontrada através da função de autocorrelação (do inglês, *Auto Correlation Function* - ACF) ou métodos como de Durbin Watson [Cohen et al., 2013].

Em alguns casos, tanto na regressão linear quanto na autoregressão, as observações de uma variável não são suficientes para encontrar de forma confiável a função que caracterize o comportamento dos dados observados em uma série temporal. Para isso, utilizam-se observações de mais variáveis na modelagem dessa função para garantir a maior acurácia. Quando isso ocorre, a regressão linear estima, além da função de y_t em x_t, representada por β, as funções entre as variáveis x_t. Isto é, a relação entre as observações das variáveis independentes também é calculada. Logo a regressão linear se apresenta como:

$$y_t = \alpha + \beta_1 x_{1t} + \beta_2 x_{2t} + \beta_3 x_{3t} + \ldots + \beta_k x_{nt} + e_{nt}$$

Onde n é a quantidade de variáveis explicativas. Com isso, a regressão linear passa a ser chamada de Regressão Linear Múltipla (RLM) .

A precisão da função encontrada está diretamente relacionada com o termo e_t, uma vez que este representa o erro do modelo [Cohen et al., 2013]. Isto é, o quanto da função encontrada em relação aos valores observados não é explicada pelos α's e β's. Alguns parâmetros são levados em consideração para determinar esta acurácia. Uma delas é a soma dos quadrados totais (SQT) que corresponde a soma dos quadrados da variável observada subraídas a média dos valores da variável dependente denotada por:

$$SQT = \sum_{t=1}^{n} (y_t - \bar{y})^2 \tag{2.1}$$

Onde $\bar{y}$ representa a média dos valores observados de y.

O SQT representa o erro máximo não explicado por uma função encontrada para representar os dados observados [Cohen et al., 2013]. Isto é, o erro neste caso representa o quão distante dos dados observados estão os pontos estimados pela média e representa o quanto da função encontrada não é explicada por α e β. Outro parâmetro é a soma dos quadrados dos resíduos (SQR) que corresponde a soma dos quadrados das diferenças entre os pontos estimados pelo modelo, e os valores observados pelo modelo encontrado, denotado por:

$$SQR = \sum_{t=1}^{n} (y_t - \hat{y})^2 \tag{2.2}$$

Onde $\hat{y}$ representa o valor estimado pelo modelo.

Este valor indica o quanto dos pontos estimados não é explicado pela função encontrada pela modelagem [Cohen et al., 2013]. O terceiro e último parâmetro avaliado é a soma dos quadrados explicados (SQE) denotada por

$$SQE = \sum_{t=1}^{n} (\hat{y}_t - \bar{y})^2 \tag{2.3}$$

Que corresponde a soma das diferenças entre os pontos estimados pelo modelo e a média dos valores já observados pela variável [Cohen et al., 2013]. Este valor representa o quanto os pontos estimados são explicados pelo modelo encontrado. Logo, dizemos que:

$$SQT = SQE + SQR \tag{2.4}$$

Em termos gerais, quanto menor o valor da SQR da função encontrada, mais o modelo se ajusta aos dados e melhor a função encontrada representa o comportamento dos dados [Cohen et al., 2013, Montgomery et al., 2006]. Esta representação é padronizada através do coeficiente de determinação, chamado de r^2 que é dado por:

$$r^2 = \frac{SQE}{SQT}$$

Logo, r^2 determina em porcentagem quanto do SQT é explicado pelo SQE, isto é, quanto da soma dos quadrados totais dos dados observados é explicado da soma da função encontrada. Para encontrar estes coeficientes (α , β, e e r^2) expostos até aqui, uma das técnicas mais utilizadas é o método dos mínimos quadrados [Montgomery et al., 2006]. Através deste método é possível

analisar e inferir relações entre as variáveis observadas. Como mencionado anteriormente, quanto menor o SQR, melhor é a função gerada pela regressão. Logo, para encontrar o menor SQR possível existem métodos que estendem a função encontrada, tratando o termo e_{nt} inicialmente encontrado como ponto de referência. Nesses métodos são executados cálculos iterativos os valores dos demais coeficientes e comparando-os com os dados observados para validar se o menor SQR foi encontrado. Um dos métodos utilizados para esta finalidade é o método do gradiente [Andrychowicz et al., 2016], que é utilizado pelo método CGP explicado na próxima subseção.

2.3.2 Processo Causal em Grafos (CGP)

O Processo Causal em Grafos (CGP) é um método que pode ser descrito como um processo autorregressivo multivariado em uma série temporal no qual seus coeficientes são filtros de grafos [Mei e Moura, 2015, Sandryhaila e Moura, 2013]. Ou seja, através de autorregressão e cálculos de coeficientes de autocorrelação, o método evidencia as interrelações entre variáveis. Neste estudo as variáveis são representadas pelos *hosts* de rede, a partir de uma única entrada. Diferente dos processos convencionais de autorregressão, o método CGP não assume a propriedade de Markov nas observações das variáveis nas séries temporais, isto é, diferente de outros processos autorregressivos, o CGP assume que todas as variáveis são dependentes. No contexto abordado neste trabalho, isso significa que o conjunto de amostras da série temporal de entrada no tempo t é influenciado de alguma forma pelo conjunto de sinais matriz de entrada no tempo $t-1$, sendo t uma intervalo de tempo pré-definida no início do processo [Mei e Moura, 2015].

Considere $x[t]$ uma série temporal na seguinte forma,

$$\begin{aligned}
\mathrm{x}[t] &= \mathrm{w}[t] + \sum_{i=1}^{p} Pi(\mathbf{A},\mathbf{c})\mathrm{x}[t-i] \\
&= \mathrm{w}[t] + \sum_{i=1}^{p}(\sum_{j=0}^{i} c_{ij}\mathbf{A}^j)\mathrm{x}[t-i] \\
&= \mathrm{w}[t] + (c_{10}\mathbf{I} + c_{11}\mathbf{A})\mathrm{x}[t-1] \\
&+ (c_{20}\mathbf{I} + c_{21}\mathbf{A} + c_{21}\mathbf{A}^2)\mathrm{x}[t-2] + \ldots \\
&+ (c_{p0}\mathbf{I} + \ \ldots \ + c_{pp}\mathbf{A}^p)\mathrm{x}[t-p]
\end{aligned} \tag{2.5}$$

Onde t é uma amostra, $Pi(\mathbf{A}, \mathbf{c})$ é um polinômio característico da matriz em $\mathbf{A}$ de ordem i; $\mathrm{w}[t]$ é ruído estatístico; c_{ij} são coeficientes polinomiais escalares, onde $\mathbf{c} = (c_{10}\ c_{11}\ \ldots\ c_{ij}\ \ldots\ c_{pp})^{(t)}$ é um vetor de todos os c_{ij}, e p é a ordem da autorregressão [Mei e Moura, 2015, Cohen et al., 2013]. Como previamente descrito na subseção 2.3.1, o CGP se enquadra dentro dos modelos autorregressivos multivariados. A modelagem da função que representa essa influência possibilita a estimativa das relações entre as variáveis observadas na série temporal de entrada. Com isso, dado um grafo $G(V, \mathbf{A})$ com um $\mathbf{A}$ *desconhecido* para estimar a estrutura da matriz de adjacência (autocorrelação)$\mathbf{A}$ das séries temporais dadas de entrada, o CGP (Equação 2.5) emprega três passos:

1. Resolver para $R_i = P_i(\mathbf{A})$, no qual $P_i(\mathbf{A})$ são coeficientes de $\mathbf{A}$ estimados através do método dos mínimos quadrados;

2. Recuperar a estrutura de **A** usando $\hat{\mathbf{A}} = \hat{\mathbf{R}}$ como em

$$\begin{aligned}\hat{\mathbf{R}}_i = \underset{\mathbf{R}_i}{argmin}\frac{1}{2}\sum_{t=p}^{t-1}\left\|\mathrm{x}[t] - \textstyle\sum_{i=1}^{p}\mathbf{R}_i\mathrm{x}[t-j]\right\|_2^2 \\ +\lambda_1\left\|\mathrm{vec}(\mathbf{R}_1)\right\|_1 + \lambda_3\sum_{j\neq i}\left\|\left[\mathbf{R}_i, \mathbf{R}_j\right]\right\|_F^2\end{aligned} \tag{2.6}$$

3. Estimar c_{ij} em uma de duas maneiras, estimando $\hat{\mathbf{c}}$ como em

$$\hat{\mathbf{c}}_i = \underset{\mathbf{c}_i}{argmin}\frac{1}{2}\left\|\mathrm{vec}(\hat{\mathbf{R}}_i) - \mathbf{Q}_i\mathbf{c}_i\right\|_2^2 + \lambda_2\left\|\mathbf{c}_i\right\|_i \tag{2.7}$$

onde

$$\mathbf{Q}_i = (\mathrm{vec}(\mathbf{I})\ \mathrm{vec}(\hat{\mathbf{A}})\ \ldots\ \mathrm{vec}(\hat{\mathbf{A}}^i)), \mathbf{c}_i = (c_{i0}c_{i1}\ \ldots\ c_{ii}). \tag{2.8}$$

Dependendo da dispersão dos dados de entrada, a otimização realizada pelo CGP utilizando projeção gradiente [Figueiredo et al., 2007] para encontrar as estimativas de $\hat{\mathbf{R}}_\mathbf{i}$ pode não convergir. Isto é, não encontrar um mínimo global ou mesmo um mínimo local. Contudo, para restringir a execução do método e tentar garantir o cálculo das estimativas, o CGP pode ser representado por um algoritmo com três passos chamado "Algoritmo Básico de Estimação" que estima $\hat{\mathbf{A}}$ e $\hat{\mathbf{c}}$ para a matriz de adjacência. No Algoritmo 1, os sobrescritos denotam o número da iteração, $\hat{\mathbf{R}}_{<i}^{(t)}$ denota $\{\hat{\mathbf{R}}_j^{(t)} : j < i\}$ assim como $\hat{\mathbf{R}}_{>i}^{(t)}$ denota $\{\hat{\mathbf{R}}_j^{(t)} : j > i\}$.

Algoritmo 1 Algoritmo Básico de Estimativa do A (CGP)

1: Inicie, $t = 0, \hat{\mathbf{R}}^{(t)} = 0$
2: **while** $\hat{\mathbf{R}}^{(t)}$ não convergir **do**
3: **for** $i = 1 : M$ **do**
4: Encontre $\hat{\mathbf{R}}^{(t)}$ com $\hat{\mathbf{R}}_{<i}^{(t)}, \hat{\mathbf{R}}_{>i}^{(t)}$ fixo usando a equação 2.6
5: **end for**
6: $t \leftarrow t + 1$
7: **end while**
8: Defina $\hat{\mathbf{A}} = \hat{\mathbf{R}}_1^{(t)}$
9: Resolva para $\hat{\mathbf{c}}$ a partir de $\mathbf{X}, \hat{\mathbf{A}}$ usando a equação 2.7

Em suma, com base nos valores de entrada considerando que as observações das variáveis nas séries temporais têm influência umas sobre as outras e sobre si mesmas, o CGP utiliza o método dos mínimos quadrados e encontra as funções que apresentam o menor erro residual das autoregressões. O CGP calcula a autoregressão de ordem 2 (ou *lag-2*) nas observações dos *hosts*, ou seja, ele utiliza duas cópias atrasadas em uma e duas amostras respectivamente das observações de cada *host* para encontrar as funções que representam o comportamento observado. Ao final, os valores dos coeficientes estimados pela funções encontradas após t iterações são utilizadas nas estimativas da autocorrelação entre essas variáveis. Essas estimativas são calculadas utilizando o cálculo da convolução entre os modelos encontrados das autoregressões [Stein e Weiss, 2016] indicando matriz de interrelações das variáveis possibilitando a inferência da causalidade delas.

Dentro do contexto deste estudo, a matriz estimada representa a matriz de interrelações entre os *hosts* da rede. Logo, os *hosts* que apresentem maior relação positiva de influência sobre os demais *hosts* podem indicar os *bots*, assim como os *hosts* que apresentem maior relação

negativa de influência sobre os demais *hosts* podem ser indicar as vítimas. Com isso, assume-se que quanto maior a similaridade na magnitude de relação entre os *hosts*, maior a coordenação. Quanto maior sua coordenação, maiores as chances de ser uma *botnet* [Mirkovic et al., 2002].

Calculando o valor dos quartis [Clarke e Cooke, 1978] da matriz de interrelações $\mathbf{X}$ resultante através do algoritmo 1 e considerando os valores acima do terceiro quartil, pode-se identificar as magnitudes que representam os comportamentos de maior influência com a maior similaridade em toda a matriz de interrelações. Sendo assim, o Algoritmo 1 toma a forma:

Algoritmo 2 CGP Adaptado para Botnets

1: Inicie, $t = 0, \hat{\mathbf{R}}^{(t)} = 0$
2: **while** $\hat{\mathbf{R}}^{(t)}$ não convergir **do**
3: **for** $i = 1 : M$ **do**
4: Encontre $\hat{\mathbf{R}}^{(t)}$ com $\hat{\mathbf{R}}^{(t)}_{<i}, \hat{\mathbf{R}}^{(t)}_{>i}$ fixo usando a equação 2.6
5: **end for**
6: $t \leftarrow t + 1$
7: **end while**
8: Defina $\hat{\mathbf{A}} = \hat{\mathbf{R}}^{(t)}_1$
9: Resolva para $\hat{\mathbf{c}}$ a partir de $\mathbf{X}, \hat{\mathbf{A}}$ usando a equação 2.7
10: Defina **limiar** = Terceiro quartil de $\mathbf{X}$
11: **for** $i = 0 : \mathbf{X}$ **do**
12: **for** $j = 0 : \mathbf{X}$ **do**
13: **if** $\mathbf{X}_{ij} <$ **limiar then**
14: $\mathbf{X} = \mathbf{X} - \mathbf{X_{ij}}$
15: **end if**
16: **end for**
17: **end for**

No que diz respeito à complexidade computacional, o CGP utiliza a uma implementação do método do gradiente chamada projeção gradiente para reconstrução de vetores dispersos (do inglês, *Gradient Projection for Sparse vector Reconstruction* - GPSR) [Figueiredo et al., 2007] onde os autores mencionam que a complexidade é no pior dos casos $O(n^2)$. Se interpretado como um processo de treinamento de um como máquina de vetores de suporte (do inglês, *Support Vector Machine* - SVM), o CGP apresenta o complexidade $O(m * n^2)$, sendo m o número de observações das n variáveis (*hosts*) da série temporal de entrada [Mei e Moura, 2015].

2.4 REVISÃO BIBLIOGRÁFICA

Na literatura, observa-se a evolução das técnicas de detecção e a identificação de *botnets* permitindo-as que sejam descritas cronologicamente e classificadas como (*i*) baseadas em assinaturas, (*ii*) detecção de anomalias e (*iii*) mineração de dados [Feily et al., 2009]. No geral, essas abordagens seguem dois procedimentos onde no primeiro momento o ataque é detectado e depois a *botnet* é identificada. Em alguns casos da literatura, as abordagens tratam somente a detecção ou somente a identificação. A Figura 2.4 ilustra genericamente essas abordagens.

As abordagens baseadas em reconhecimento de assinaturas são as mais antigas e imitam no primeiro momento a intuição humana. Essas abordagens utilizam uma base de conhecimento que contém dados correspondentes aos comportamentos e características de *botnets* conhecidas para serem comparadas com os fluxos da rede a fim de detectar atividade

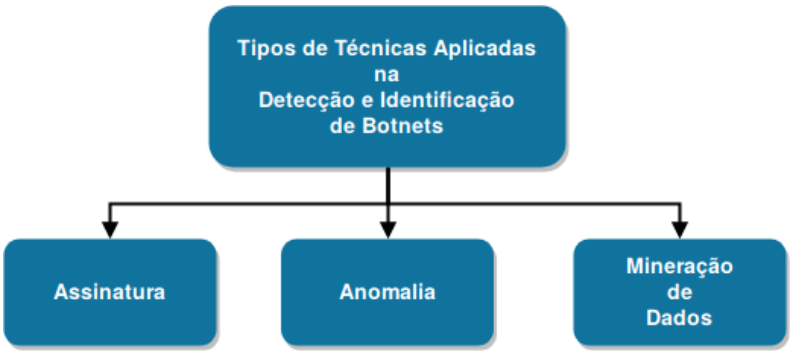


Figura 2.4: Abordagens de Detecção e Identificação de Botnets na Literatura

de *botnet* [Caswell et al., 2003], por isso, elas dependem de uma base de dados para funcionar. As abordagens baseadas em detecção de anomalias podem ser descritas como uma evolução das abordagens baseadas em assinatura, porque apesar de mais complexas, não deixam de se comportar como as técnicas baseadas em assinatura. As técnicas baseadas em detecção de anomalia tomam como referência um comportamento considerado normal da rede. Esse comportamento normal é determinado seja através de seu valor de entropia ou aprendizado de máquina, onde qualquer variação nesse comportamento (anomalia) serve de gatilho para o processo de identificação da *botnet* e posterior acionamento de contramedidas [Binkley e Singh, 2006, Karasaridis et al., 2007, Gu et al., 2008b, Ahmad et al., 2017]. Ainda dentro das técnicas de detecção e identificação baseadas em anomalia, abordagens baseadas em tráfego DNS são as únicas que apresentam a possibilidade de detecção e identificação de forma online *botnet* [Schonewille e van Helmond, 2006, Ramachandran et al., 2006, Choi et al., 2007].

As abordagens baseadas em mineração de dados são mais sofisticadas que as baseadas em assinatura e em anomalia, porém demandam maior poder computacional. Ainda assim, em alguns estágios das abordagens baseadas em mineração de dados é possível observar características das abordagens baseadas em assinaturas e em anomalias, como a definição e identificação de padrões e a detecção de dados anômalos. Nas abordagens baseadas em mineração de dados, é realizada a análise dos registros de rede como um todo, normalmente após a detecção do ataque ou em uma infraestrutura intermediária que utiliza técnicas de mineração de dados como associação, agrupamento (clusterização), classificação e categorização para detectar e identificar as atividades de uma *botnet* na rede [Goebel e Holz, 2007, Strayer et al., 2008, Masud et al., 2008, Gu et al., 2008a, Lagraa et al., 2017, Kong et al., 2016, Guerrero et al., 2017]. Essas abordagens dependem fortemente de um estágio de treinamento e consideram um estado predeterminado da rede para realizar a detecção e a identificação das *botnets*. O método avaliado neste trabalho se enquadra nas abordagens de mineração de dados, com uma característica particular de possibilitar a detecção e a identificação em um mesmo processo, permitindo sua aplicação de forma contínua no mecanismo de defesa da rede sem um processo de treinamento específico como nas demais abordagens de mineração de dados.

A Tabela 2.1 apresenta um comparativo das técnicas de detecção de *botnets* presentes na literatura abordando as características como a capacidade de detecção de *botnets* desconhecidas, dependência de protocolo, detecção de *bots* encriptados, detecção online, e baixa taxa de falso positivo. Em seguida são apresentadas as particularidades dos trabalhos que se enquadram nas abordagens de mineração de dados e suas relações e diferenças com o método CGP avaliado neste trabalho. As técnicas apresentadas pelos trabalhos classificados na Tabela 2.1 não estão dispostas em ordem cronológica de publicação, contudo observa-se que existe uma evolução ao longo do tempo em respeito aos métodos e às técnicas de detecção. Os trabalhos mais antigos concentram

as técnicas de detecção baseadas em assinatura, seguida das técnicas baseadas em anomalia e das técnicas que abordam o tráfego DNS e as abordagens mais recentes concentram as técnicas de mineração de dados. Dado que o CGP se enquadra dentro das técnicas de mineração de dados, o restante deste capítulo detalha em ordem cronológica algumas técnicas também baseadas em mineração de dados que apresentam alguma similaridade ou associação com o método CGP adaptado e avaliado neste trabalho.

Embasamento da Técnica	Técnicas de Detecção	Detecção de Botnets Desconhecidos	Independente de Estrutura e Protocolo	Detecção de Botnets Encriptadas	Detecção Online	Baixa Taxa de Falso Positivo
Assinatura	[Caswell et al., 2003]	x	x	x	x	x
Anomalia	[Binkley e Singh, 2006]	✓	x	x	x	x
	[Karasaridis et al., 2007]	✓	x	✓	x	✓
	[Gu et al., 2008b]	✓	x	✓	x	✓
	[Ahmad et al., 2017]	✓	✓	✓	x	x
Tráfego DNS	[Schonewille e van Helmond, 2006]	✓	x	✓	x	✓
	[Ramachandran et al., 2006]	✓	x	✓	✓	x
	[Choi et al., 2007]	✓	✓	✓	x	✓
Mineração de Dados	[Goebel e Holz, 2007]	✓	x	x	x	x
	[Strayer et al., 2008]	✓	x	x	x	x
	[Masud et al., 2008]	✓	✓	✓	x	✓
	[Gu et al., 2008a]	✓	✓	✓	x	✓
	[Kong et al., 2016]	✓	✓	✓	x	x
	[Lagraa et al., 2017]	✓	✓	✓	x	✓
	[Guerrero et al., 2017]	✓	✓	✓	x	x

Tabela 2.1: Comparativo entre Métodos de Detecção de Botnets

Em 2008, [Masud et al., 2008] propuseram uma técnica para detectar *botnets* geradoras de DDoS baseado-se na identificação de mensagens de comando e controle (C&C) nos registros de tráfego da rede local. Com essa técnica as mensagens trocadas entre *hosts* são analisadas e correlacionadas durante um período predeterminado. Este processo permite que os fluxos de rede sejam categorizados de acordo com sua similaridade. Essa similaridade auxilia no treinamento de um algoritmo de classificação que divide os tipos de fluxos em basicamente três categorias. A primeira categoria é a *Bot-response* que é caracterizada pela resposta dada por um *bot* ao *botmaster*. A segunda categoria é a *Bot-app* que é caracterizada pelo envio de um comando do *botmaster* para o *bot*, podendo ser um comando de ataque ou de requisição de informações dos *hosts* infectados. A terceira e última categoria é a *Bot-other* que é caracterizada pelo tráfego originado de um possível *bot* para outros *hosts* que não tenham o comportamento similar, o que pode indicar um ataque. Essa ténica utiliza dez características dos fluxos de rede para auxiliar na classificação dos *bots*. O conjunto dessas características é escolhido com base na premissa de que o tempo de requisição e resposta entre *bots* é constante e mais rápido que o humano, facilitando a identificação da *botnet*. Nesse trabalho foram utilizadas as técnicas de máquina de vetores de suporte (do inglês, *Support Vector Machine* - SVM), árvore de decisão (do inglês, *Decision Tree* - DT) e Naive Bayes para realizar a classficação que de acordo com seus autores, obtiveram 98% de acurácia da identificação dos *bots*, sem falsos positivos e uma máxima de 2% de falsos negativos. A principal diferença dessa técnica para a abordada neste trabalho é a quantidade de características utilizadas, onde o CGP utiliza duas características enquanto a técnica acima utiliza dez.

Também em 2008, [Gu et al., 2008a] propuseram o BotMiner, um *framework* que utiliza técnicas de mineração de dados para detectar tráfego C&C na rede. Este *framework* é um aprimoramento do *BotSniffer* [Gu et al., 2008b]. Assim como a técnica mencionada no parágrafo anterior, esta agrupa os fluxos de rede de acordo com a similaridade de atividade. Para isso esse *framework* divide as atividades em três módulos principais. O primeiro módulo é chamado de

A-Plane, que é responsável por detectar e registrar atividades maliciosas na rede como *port scan*, *spamming*, e tentativa de exploração de vulnerabilidades conhecidas Internet. O segundo módulo é chamado de *C-Plane* e é responsável por criar um registro de fluxo da rede local utilizando diversas características da rede de forma que facilite uma posterior análise desses fluxos. Ambos os módulos mencionados até o momento agrupam os hosts de acordo com suas atividades e fluxos para posterior análise. O terceiro módulo é chamado de *Cross-Plane Correlator* e é responsável por correlacionar os agrupamentos gerados pelo *A-Plane* e pelo *C-Plane* classificando assim os possíveis *bots* na rede local. De acordo com os autores, o BotMiner apresentou uma acurácia média de 96%, onde a taxa de falso positivo foi menor que 0,003%. Contudo os próprios autores afirmam que o tempo e a quantidade de recursos computacionais necessários para que a detecção e identificação pode tornar a técnica ineficaz. A principal diferença do CGP para o BotMiner além da quantidade de características utilizadas, é a quantidade de dados da rede analisados para a detecção e identificação da *botnet*.

Em 2011, [Thapngam et al., 2011] propuseram o uso da autocorrelação de Pearson em séries temporais considerando a taxa recepção de pacotes para diferenciar um ataque de negação de serviço de um aumento da vazão da rede causado por usuários legítimos. Para isso, as características utilizadas são pares compostos por endereço de origem e porta dos *hosts* da rede associados ao tráfego gerado por estes pares. Nesse estudo foi comprovado que utilizando essa abordagem é possível diferenciar esses dois tipos de tráfego. Contudo, os autores não abordam a origem dos ataques, isto é, o método proposto é capaz de informar ao administrador da rede se o aumento do volume de tráfego e a possível negação de serviço é causada pelo aumento repentino de usuários legítimos associado ao incompatível dimensionamento da rede, ou por um ataque intencional gerado *bots*. Além disso, os autores não abordam a identificação dos *bots*.

Em 2016, [Kong et al., 2016] propuseram uma estrutura de agrupamento para detectar *botnets* usando uma otimização do algoritmo de agrupamento árvore hierárquica euclidiana geradora mínima (do inglês, *Hierarchical Euclidean Minimum Spanning Tree* - HEMST). Essa técnica utiliza uma série de características genéricas de rede, como tamanho de pacote, tipo de protocolo e espaçamento de segmento. O valor desses atributos são extraídos da captura de rede como parâmetros para criar agrupamentos de *hosts* baseado na similaridade dos fluxos de dados. Esses agrupamentos então são utilizados para facilitar o processo de identificação de *bots* após o início de um ataque DDoS por exemplo. Os resultados deste trabalho apontam que essa técnica é mais rápida e eficaz se comparada com os métodos de agrupamento utilizados na deteção e identificação de *botnets* até então. Contudo, os autores mencionam que esses agrupamentos só são úteis para detectar e identificar uma *botnet* após a detecção do ataque DDoS. A quantidade de características utilizadas para a criação dos agrupamentos é a principal diferença entre o método estudado neste trabalho e os demais métodos mencionados até o momento.

Em 2017, [Lagraa et al., 2017] propuseram o BotGM, uma técnica de mineração de dados em grafos não supervisionado que detecta e identifica a origem de *botnets* utilizando os fluxos de rede considerando os endereços e portas de origem e destino dentro de uma janela de tempo (chamados pelos autores do trabalho de eventos) para estabelecer uma correlação entre essas características. Quando ocorrem eventos considerados atípicos na rede, estes são classificados como eventos gerados por um *bot* e portanto, quando todo um conjunto de eventos atípicos são detectados, uma *botnet* é exposta. Nesse trabalho o agrupamento de *hosts* é utilizado para classificação dos *bots* e portanto é comparado apenas com os métodos que utilizam abordagens similares. O resultado obtido pelos autores desse trabalho foi de até 95% de acurácia na detecção de *botnets*. Contudo, os autores ressaltam que o método pode ser menos eficaz que outros que utilizam análises individuais dos fluxos da rede, o que ainda assim não inviabiliza

o BotGM. O conceito utilizado por essa abordagem é similar ao utilizado neste estudo, onde baseado na observação do comportamento dos fluxos de rede é possível identificar a *botnet*.

Uma das características em comum entre todos os trabalhos mencionados acima é a utilização de diversos parâmetros e características da rede para a detecção do ataque DDoS e posterior identificação da *botnet*. Na maioria deles, existe a premissa de que os dados avaliados estejam normalizados e que a seleção do conjunto de características utilizadas para a detecção de ataques de negação de serviço acompanhe tipos específicos de DDoS. Isto é, cada configuração realizada pelas técnicas abordadas prevê primeiro a detecção de um ataque DDoS para posterior identificação de um tipo específico de *botnet*, seja ela conhecida ou não. Isso significa que quando o comportamento da *botnet* é conhecido, a detecção e a identificação são realizadas de forma contínua e eficaz, contudo, quando a *botnet* é desconhecida, o processo de identificação da *botnet* é direcionado a um terceiro mecanismo, que indica a possível *botnet*. Essas abordagens são arriscadas, pois não consideram a diferenciação de um ataque DDoS de um evento de *flash-crowd* que consiste no aumento repentino de tráfego na rede causado usuários legítimos buscando por exemplo, promoções e descontos em sites de vendas. Uma vez que todos os testes foram realizados com conjuntos de dados conhecidos por serem apenas de ataques DDoS como o [Hick, 2013] e [García et al., 2014]. Outro ponto ausente nestes trabalhos é a validação das técnicas em conjuntos de dados que não apresentam ataques DDoS, o que pode alterar os resultados expostos por estes métodos.

Acompanhando os objetivos das técnicas detalhadas acima, este trabalho estuda a eficácia do método CGP como método de detecção e identificação de *bots* através da similaridade da emissão de pacotes na rede [Mirkovic et al., 2002] sem a necessidade de um estágio de treinamento e de aprendizado sobre a rede legítima ou a *botnet*. Isso porque o método CGP permite a modelagem das interrelações entre elementos através de cálculos autorregressivos sobre uma série temporal. Esta série temporal é composta por amostras de observações destes elementos em um intervalo de tempo predeterminado. Uma das principais vantagens do método CGP em relação aos demais é a utilização de apenas uma característica do tráfego de rede para detectar e identificar a *botnet* com uma baixa taxa de falso positivo.

2.5 RESUMO

Este capítulo apresentou a descrição e crítica dos trabalhos relacionados ao problema de detecção e identificação de botnets, tratado neste trabalho. Além disso, são revistos os fundamentos necessários para contextualizar o problema e o estudo realizado, abordando os conceitos básicos de redes e ataques de negação de serviço distribuído (do inglês, *Distributed Denial of Service* – DDoS). Em seguida, foram apresentados os conceitos que fundamentam a análise e o processamento dos dados e, por consequência, o método de Processamento Causal em Grafos (do inglês, *Causal Graph Process* – CGP), que é objeto de estudo deste trabalho em relação à sua aplicação para detectat e identificar botnets. No próximo capítulo, é descrita a metodoglogia de avaliação empregada na avaliação da viabilidade do uso do método CGP na detecção e identificação de *botnets*.

3 MÉTODO DE DETECÇÃO E IDENTIFICAÇÃO DE BOTNETS

Este capítulo apresenta o método para a detecção e a identificação de *botnets* geradoras de ataques DDoS. Como fundamentado no capítulo anterior, o método de detecção toma como base o método do Causal Graph Process (CGP). O CGP realiza uma autorregressão em séries temporais, para estimar uma rede de interações e indicar o grau de influências entre os *hosts*. No contexto deste estudo, assume-se que a magnitude das interações representadas pela matriz de adjacência resultante do método é diretamente proporcional à coordenação entre os *bots* ativos durante um ataque DDoS. O objetivo é verificar a eficácia do método CGP na detecção e identificação de *botnets* gerdoras de DDoS volumétrico. Este capítulo está dividido em três partes. A Seção 3.1 apresenta uma visão geral do cenário de avaliação e suas características, considerando onde o método CGP pode ser posicionado na rede. A Seção 3.2 descreve os passos realizados para a avaliação do método CGP, detalhando os procedimentos realizados. A Seção 3.3 resume o capítulo.

3.1 VISÃO GERAL

A adaptação e eficácia da utilização do método CGP é avaliada neste trabalho considerando seu uso para a detecção e a identificação de *botnets* de forma contínua. A adaptação é necessária porque o CGP é um método genérico que estima as interrelações entre variáveis. Isto é, o CGP não foi projetado especificamente para detectar e identificar *botnets*, essa aplicação específica é desenvolvida neste estudo. A eficácia da adaptação do CGP corresponde à quantidade de dispositivos identificados como *bot* comparando com outros métodos. O funcionamento do CGP se difere dos demais métodos por conter a processos que diferem dos demais métodos baseados em mineração de dados que requerem um estagio de treinamento. A não necessidade de um estágio de treinamento se dá porque o CGP utiliza observações anteriores das próprias variáveis para inferir as relações entre elas.

O método CGP assume a existência de características relacionais entre variáveis que implicam na correlação e em alguns casos na causalidade entre elas. No contexto deste estudo, a adaptação do CGP se dá através da interpretação das variáveis como sendo os *hosts* de uma rede, a característica relacional como sendo a capacidade de emitir pacotes pelos *hosts*, e as observações sendo a quantidade de pacotes emitidas pelos *hosts* em determinado intervalo de tempo. Estas observações estão dispostas como uma série temporal que é processada por uma implementação do método que retorna uma matriz de interrelações entre os *hosts* observados. Os valores da matriz de interrelações estão dispostos como uma matriz de adjacência onde assumimos que magnitude dos valores das interrelações são diretamente proporcionais à coordenação entre os *hosts* no período avaliado. Isto é, assim como nos cálculos de coeficiente de correlação entre variáveis, o quanto maior a similaridade entre as magnitudes dos valores das variáveis na matriz de interrelações, mais os *hosts* estão coordenados dentro do período avaliado. Isso torna o método CGP indiferente a possíveis mudanças na estrutura da rede defendida, uma vez que esta série temporal corresponde ao tráfego da rede no período registrado, tornando o método adequado para a detecção e identificação das diferentes topologias de *botnets*. O objetivo é que o método possa atuar tanto no roteador, seja ele de borda ou núcleo, quanto no *firewall*. A Figura 3.1 ilustra de forma genérica um o posicionamento de um dispositivo na rede local contendo uma implementação método para detecção e identificação de *botnets* geradoras de DDoS.

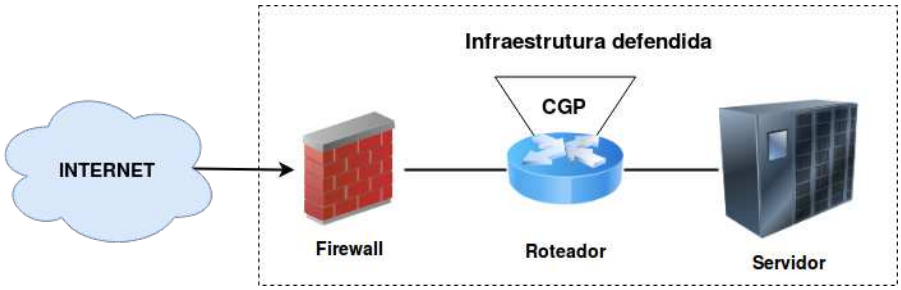


Figura 3.1: Posicionamento do CGP no Roteador de Borda

Como mencionado anteriormente e observado na Figura 3.1, o método CGP pode atuar tanto no sistema de *firewall* quanto no roteador, seja ele de borda ou núcleo. Na Figura 3.1 os componentes da infraestrutura estão dispostos de uma maneira genérica a fim de ilustrar o exemplo de posicionamento de atuação do método. Na figura, o método CGP está posto no roteador de borda da infraestrutura defendida, fazendo com que o método atue como o agente responsável pela identificação de *botnets* em toda a rede gerenciada pelo roteador de borda. O método pode ser implementado também dentro dos roteadores de núcleo, onde a detecção das *botnets* é realizada nas sub-redes da infraestrutura defendida. Desta forma o roteador de borda não assume toda a responsabilidade na detecção e identificação das *botnets*, delegando aos roteadores de núcleo esta função. Quando implementado dentro do sistema de *firewall*, o método CGP pode se apresentar dentro de um dispositivo dedicado, como um microcomputador com mais recursos de memória e armazenamento que os roteadores. Deste modo, o método não onera os recursos dos roteadores contribuindo com a manutenção e coesão da rede. A Figura 3.2 ilustra o posicionamento do método dentro do sistema de *firewall*.

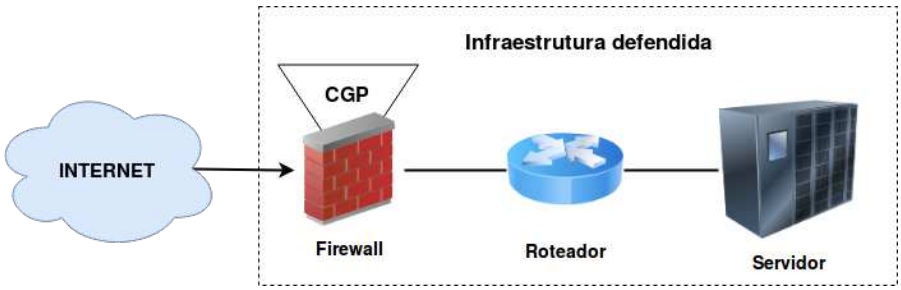


Figura 3.2: Posicionamento do CGP no *Firewall*

3.2 CONFIGURAÇÃO DO CENÁRIO

Esta seção apresenta a metodologia e a avaliação do uso do método CGP na detecção e na identificação de *botnets*, assim como a descrição e o funcionamento de cada um dos passos e procedimentos.

De forma geral, a avaliação é composta por dois passos. O primeiro corresponde a adaptação do CGP para o uso em redes de computadores, onde os são realizados procedimentos para que que os dados observados dos *hosts* de uma rede possam ser processados pelo método e contém os procedimentos responsáveis pela extração e manipulação dos dados assim como os procedimentos para a execução de uma implementação do CGP. O segundo corresponde aos métodos utilizados neste trabalho para avaliar a eficácia do CGP como método de detecção e identificação de *botnets* onde estão descritos os procedimentos para a análise dos dados manipulados no primeiro passo. O primeiro passo segue os procedimentos (i) Extração e

Formatação dos Dados, (*ii*) Cálculo de influências utilizando o método CGP. O segundo passo segue a (*i*) Análise da matriz de influências gerada pelo método e a (*ii*) Tomada de ação pelo sistema de defesa. A Figura 3.3 ilustra a arquitetura geral da avaliação do método.

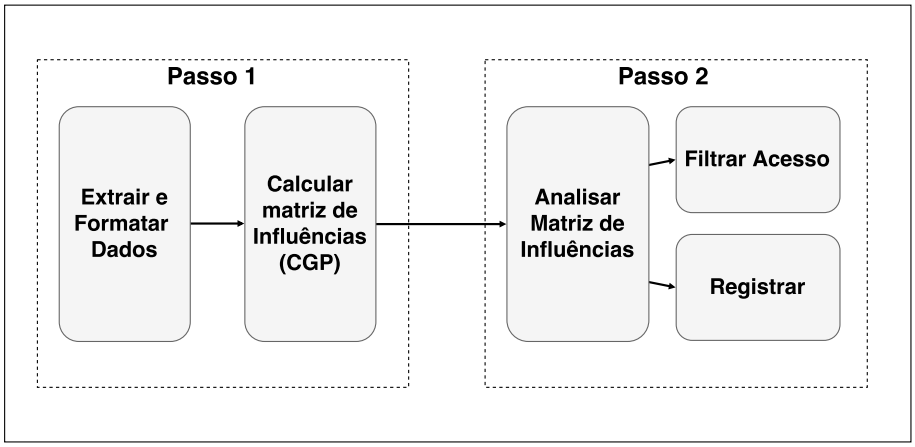


Figura 3.3: Metodologia de Avaliação

3.2.0.1 Passo 1

O procedimento *Extração e Formatação dos Dados* tem como finalidade monitorar continuamente o fluxo de dados da rede registrando-o em memória através de um arquivo formatado como uma série temporal contendo dados observados dos *hosts* detectados na captura por determinado período. Este período é definido pelo administrador da rede e corresponde ao conjunto de observações a serem processadas pelo método CGP. O intervalo entre as observações também é definido pelo administrador da rede e determina a precisão na classificação da *botnet* e da avaliação do uso do método na detecção e na identificação de *botnets* geradoras de DDoS. Os dados observados são a soma da quantidade de pacotes ou a soma do tamanho dos pacotes por intervalo. O *cálculo de influências* é a aplicação da implementação do Algoritmo 1 na série temporal gerada no componente (*i*) que tem como saída uma matriz $N \times N$, em que N são os *hosts* presentes na série temporal dada como entrada. Os valores indexados correspondem à magnitude de interrelações entre os *hosts*. Nessa avaliação assumimos que a similaridade na magnitude de interrelações entre os *hosts* representa também a coordenação entre eles. A Figura 3.4 ilustra o processo realizado no passo 1.

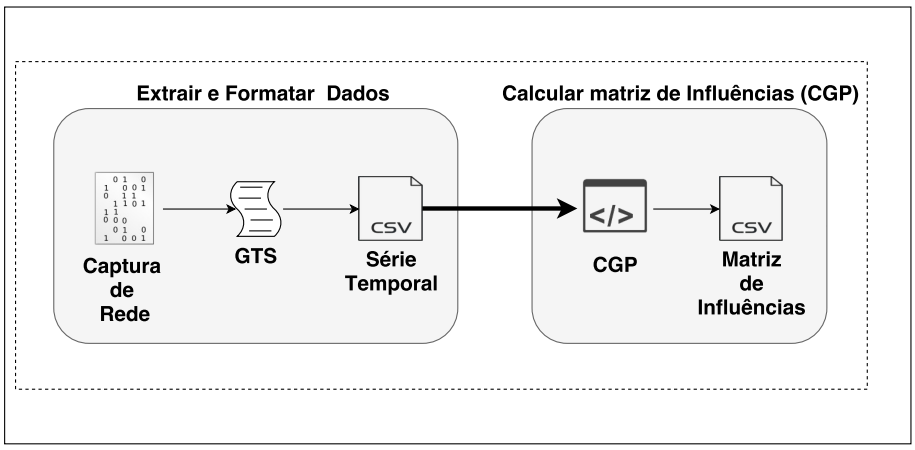


Figura 3.4: Passo 1: Extração, Formatação e Cálculo da Matriz de Influências

Como ilustrado na Figura 3.4, a captura do tráfego de rede no formato PCAP é processada por um script em SHELL chamado gerar série temporal (do inglês, *Generate Time Series* - GTS) que extrai os dados conforme segue. (*i*) a ferramenta TCPDUMP lê o arquivo de captura e armazena todas as informações em um arquivo auxiliar. (*ii*) o comando GREP e UNIQUE do *shell* identificam todos os endereços de origem dos pacotes na captura e os armazena em outro arquivo auxiliar contendo apenas os endereços únicos. (*iii*) Para cada endereço identificado, realiza-se uma varredura no arquivo auxiliar contendo as informações da captura considerando o intervalo de amostra configurado pelo administrador de rede. Este processo gera três séries temporais formatadas em um arquivo com valores separados por vírgulas (do inglês, *Comma Separated Values* - CSV), onde as informações contidas são: amostra de tempo, endereço de origem, quantidade de pacotes transmitidos pelo endereço de origem durante o intervalo, soma do tamanho dos pacotes transmitidos pelo endereço de origem durante o intervalo. Este arquivo contendo as duas séries temporais juntas pode ser utilizado para dois fins, o primeiro é a avaliação geral e seleção do período a ser processado, o segundo é a geração da série temporal no formato de matriz utilizada como entrada para o método CGP. O método então utiliza cada observação de todos os *hosts* (representados por seus endereços de origem) em todas as amostras registradas na série temporal, e efetua sobre estas a projeção gradiente a fim de encontrar os modelos com melhor ajuste aos dados observados. O algoritmo gera então uma matriz $N \times N$ onde cada N representa um *host* da rede e os valores das células são os valores estimados das interrelações entre estes *hosts* no período registrado pela série temporal. As interrelações são estimadas a partir dos modelos obtidos nas autoregressões onde a convolução entre os modelos encontrados resulta em uma terceira função que estima a relação entre os dois modelos, atribuindo valores a matriz de interrelações entre os *hosts*. Esta matriz de interrelações é então processada pelo Passo 2 onde é analisada e tratada para detecção e identificação da *botnet*.

3.2.0.2 Passo 2

O resultado do *Cálculo de Influências* é analisado de forma que os *hosts* apresentando maior similaridade nas interrelações em toda a matriz de interrelações, sejam destacados e identificados como *bots*. Essa identificação é feita através do estabelecimento de um limiar. Este limiar é obtido através do cálculo dos quartis dos valores dessa matriz onde os valores acima do terceiro quartil (correspondentes aos 25% superiores em toda a matriz) são considerados interrelações entre *bots* e a *vítima*. Utilizamos esta forma para identificar os *bots* na matriz por assumir que quanto maior a similaridade na magnitude de influência, maior a coordenação entre os *hosts*, que por sua vez assumimos ser a coordenação da *botnet* e também, por ser uma maneira eficaz de organizar e separar os dados resultantes do método CGP. Após essa identificação, as informações dos *dispositivos* identificados como *bots* podem ser enviadas para o administrador de rede ou um sistema de filtragem de pacotes no formato de arquivo de texto contendo os endereços de origem. Em seguida, o sistema de filtragem de pacotes, componente de um sistema de *firewall* que tem por finalidade limitar, redirecionar ou bloquear o tráfego de pacotes de *hosts*, executa uma das duas ações garantindo o funcionamento da rede. No caso abordado neste trabalho os *hosts* que têm seus pacotes filtrados são aqueles identificados como *bot*. A Figura 3.5 ilustra o processo realizado no Passo 2.

Como ilustrado na Figura 3.5, a matriz de influências gerada no Passo 1 é analisada utilizando a ferramenta R, onde são calculados os valores dos quartis da matriz. Depois é estabelecido um limiar considerando somente os índices que apresentam valores correspondentes aos 25% superiores (terceiro quartil) em toda a matriz de interrelações, excluindo assim os índices que por esta avaliação são considerados de influência normal ou irrelevante. Ao fazer isso, a matriz resultante é a matriz de interrelações correspondente aos *hosts* mais influentes

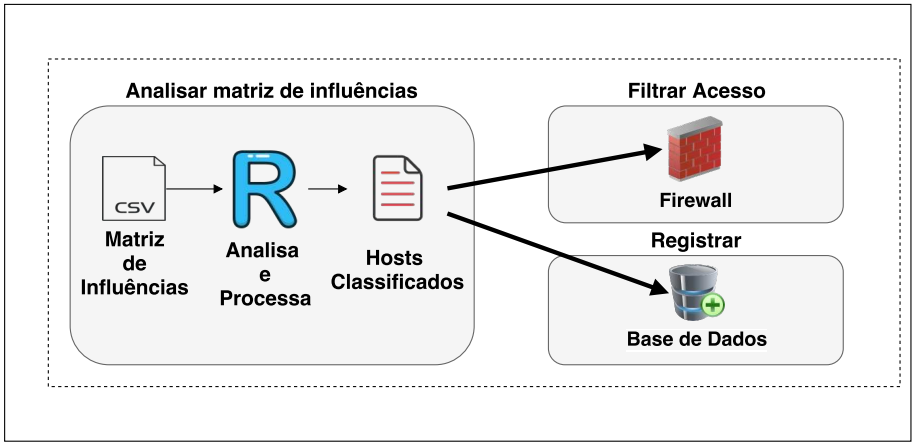


Figura 3.5: Passo 2: Análise da Matriz de Influências e Tomada de Contramedidas

da rede no intervalo de tempo avaliado, que para o cenário avaliado corresponde à *botnet*. As informações dos índices são então enviadas através de um arquivo contendo exclusivamente os endereços de origem identificados como *bots* pela análise realizada. Este arquivo é então utilizado por um sistema de filtro de tráfego para tomar as ações necessárias e um arquivo de registro (log) é gerado para auditoria. Sendo assim, salvo o procedimento do cáclulo da matriz de interrelações do Passo 1, os demais procedimentos fazem parte do processo de adaptação do CGP para a detecção de *botnets* geradoras de DDoS e são o cerne das contribuições deste trabalho.

3.3 RESUMO

Este capítulo apresentou a metodologia de avaliação sobre a utilização do método CGP na detecção e identificação de *botnets* geradoras de ataque DDoS. Este capítulo também descreve a adaptação do método CGP para redes de computadores e os possíveis posicionamentos do método CGP no mecanismo de defesa da rede, detalhando especificidades que permitem sua aplicação na detecção e identificação de uma *botnet*. Assim, foi apresentado seu funcionamento, abordando o formato dos dados de entrada, o processamento realizado e o arquivo de saída. Foi também descrito como o arquivo de saída é interpretado na identificação de uma *botnet* utilizando ferramentas de análise de dados e também possíveis ações a serem tomadas. O capítulo seguinte apresenta os resultados sobre o uso do método CGP sobre quatro cenários de ataques DDoS volumétricos e um cenário sem ataque, este último para fins de comparação.

4 RESULTADOS

Este capítulo apresenta os resultados referentes à análise método CGP na detecção e na identificação de *botnets* geradoras de DDoS. A análise segue uma abordagem orientada a traços. Desta forma, além de descrever as principais características dos cenários de avaliação, são descritos os *datasets* utilizados e o processo de escolha dos mesmos. É descrito também o tratamento realizado para padronizar os *datasets* de forma que apresentem os mesmos tipos de dados a fim de aumentar a confiabilidade e reprodutibilidade das avaliações. Além disso, descreve-se a forma como os dados foram tratados e quais as ferramentas utilizadas para extração das características utilizadas pelo método CGP. Para isso, este capítulo está dividido da seguinte forma. A Seção 4.1 descreve as bases de dados utilizadas, os processos e os parâmetros para a aplicação do método CGP. A Seção 4.2 apresenta os resultados obtidos da avaliação do uso do método CGP aplicado em cada cenário descrito nas base de dados.

4.1 BASES DE DADOS

Para avaliar a adaptação e eficácia do uso do método CGP na identificação de *botnets*, utilizou-se um grupo de *datasets* que apresentam em seus registros ataques DDoS. Dentre os *datasets* avaliados foram selecionados os conjuntos do Centro para Análise Aplicada de Dados da Internet (do inglês, *Center for Applied Internet Data Analysis* - CAIDA), da Universidade Tecnológica Tcheca (do inglês, *Czech Tecnical University* - CTU)e por uma base de dados contendo um ataque DDoS real sofrido no início do ano por uma empresa de hospedagem de serviços Secure Linux Solutions (SLS) que está intitulado neste trabalho como "SLS". A motivação para a seleção dos *datasets* mencionados se deve ao fato de apresentarem registros rotulados de ataques DDoS. Isto permite a comparação entre a *botnet* detectada e identificada com o método CGP e a *botnet* descrita nesses arquivos de descrição destes registros. Os registros estão em formato *pcap* que é um formato utilizado por diversas ferramentas de análise de tráfego em redes de computadores e que estão disponíveis na Internet. Isto permite a reprodução dos resultados. Foi também utilizada uma captura de tráfego de rede onde não há ataque DDoS sendo realizado. A captura foi realizada em um hotel em Campos do Jordão durante um evento em que um artigo relacionado a este trabalho foi publicado e está intitulado "Hotel".

Utilizaram-se neste estudo as ferramentas TCPDUMP, EDITCAP, MERGECAP, em conjunto com *scripts* em SHELL para o tratamento e extração das características e dados necessários para compor as séries temporais de entrada para o método CGP. Uma implementação em Matlab do Algoritmo 1, disponibilizada pelos autores do método CGP foi utilizada para calcular a matriz de interrelalções avaliada neste trabalho. E por fim, a ferramenta R foi utilizada para o tratamento dos dados da matriz de interrelações, geração dos dados de saída e avaliação dos resultados do método CGP. A Tabela 4.1 apresenta as especificações do computador usado para a realização dos testes e avaliação.

4.1.0.1 Características das bases de dados

"DDoS Attack 2007" é disponibilizado pelo CAIDA [Hick, 2013]. Ele contém aproximadamente uma hora de fluxo de dados coletados da rede local. Os dados estão distribuídos em três subconjuntos, all-victim, to-victim e from-victim. Sendo que o subconjunto selecionado dessa base de dados foi o all-victim. O all-victim apresenta registros de fluxos de pacote direcionadas

Componente	Especificação
CPU	Intel Core i7-3632QM (Quad core)
Memória	8GB DDR3 1600MHz
Rede	Realtek RTL811/8168/8411 Gigabit Ethernet Controller
S.O.	Debian GNU/Linux (kernel 4.9.0-6)

Tabela 4.1: Especificações Técnicas do Computador Utilizado nos Testes

da rede para vítima e da vítima para a rede. O to-victim apresenta apenas os registros de fluxo de rede direcionados para a vítima. O from-victim apresenta apenas os registros de fluxo de rede direcionada da vítima para a rede. O tamanho total do arquivo é de 21GB. De acordo com a documentação, o ataque tem início por volta das 21:13:00 UTC (19:13:00 BRST), quando a carga de rede aumenta em poucos minutos de uma taxa perto de 200 kbits/s para cerca de 80 Mbits/s. O tamanho dos pacotes nessa base de dados oscila entre 48 bytes até 1500 bytes.

CTU-13 é o nome dado a base de dados de tráfego de *botnet* capturado na CTU *(Czech Technical University)* localizado na República Tcheca, em 2011 [García et al., 2014]. A base de dados CTU-13 consiste em treze capturas distintas registradas no formato *pcap*, chamadas pelos autores de cenários, de diferentes tipos de ataques realizados por *botnets*. Em cada cenário foram executados malwares distintos, como o Neris, Rbot Virut, Menti, Sogou, Murlo, NSIS.ay e Virut. que usam protocolos como IRC, P2P e HTTP e utilizam técnicas de SPAM, ClickFraud, FastFlush e vulnerabilidades no HTTP para se espalhar e infectar outros dispositivos. Neste trabalho apenas as bases de dados que apresentam em sua descrição ataques DDoS e que possuem mais de um *bot* foram selecionados. A Tabela 4.2 apresenta detalhes sobre cenários, como a duração de cada cenário, o número de pacotes trafegados no período, o número de fluxo de dados no período, o tamanho dos arquivos de registro dos cenários, o malware utilizado e a quantidade de *bots*.

Cenário	Duração (Hours)	# Pacotes	# Fluxos	Tamanho	Bot	# Bots
4	4.21	62,089,135	1,121,077	53GB	RBot	1
10	4.75	90,389.782	1,309.792	73GB	RBot	10
11	0.26	6,337.202	107.252	5.2GB	RBot	3

Tabela 4.2: Cenários Avaliados

Como observado na Tabela 4.2, somente os cenários 10 e 11 apresentam mais de um bot, configurando uma *botnet* e logo, sendo compatível para nossa avaliação. O malware Rbot é um *bot* escrito na linguagem Ruby feito para o IRC que permite ao atacante executar através do *BotMaster* uma série de ações, variando de uma simples autenticação SSH até um ataque DDoS. Nessa base de dados, o Rbot foi alterado e compilado de forma seus autores tivessem total controle sobre o *bot*, a fim de não comprometer o funcionamento das redes ou sub-redes que não fazem parte da simulação. A versão do Rbot utilizada para a execução das simulações pode ser encontrada na página da base de dados CTU-13 em [García, 2011].

SLS é o nome dado ao conjunto de dados disponibilizado por [Fonseca, 2018]. Ele contém aproximadamente duas horas de fluxo de dados coletados na rede. A captura está no formato NetFlow. Similar ao formato PCAP, este formato armazena dentre outros dados, informações do cabeçalho dos pacotes trafegados como data e hora da emissão de pacotes, endereço IP de origem e destino dos pacotes, tamanho dos pacotes, quantidade de pacotes, etc. O tamanho total do arquivo é de aproximadamente 5GB de dados. De acordo com o autor que disponibilizou a captura, o ataque foi direcionado a um *host* que apresentava link com acesso à

Internet limitado a 50MB/s e que não era servidor, ou seja não disponibilizava nenhum serviço específico para aplicações finais, logo o ataque impactou primariamente a velocidade média de navegação da rede. Foi disponibilizado também informação sobre os *hosts* que a princípio não fizeram parte do ataque por serem constantemente monitorados pelo administrador da rede.

Hotel é o nome dado ao registro de tráfego no formato PCAP capturado nas dependências de um hotel localizado em Campos do Jordão. A captura contém aproximadamente duas horas de registro de tráfego de rede que, a princípio, não apresenta ataques DDoS. Esta captura foi realizada para validar o método CGP quando aplicado sobre os registros de uma rede sem ataque, o resultado não apresentou interrelações entre os *hosts*.

4.1.0.2 Extração das séries temporais

Antes iniciar o Passo 1 da avaliação do CGP, fez-se necessário padronizar os tipos de informações contidas em cada um das bases de dados. Isto porque que na base de dados da CAIDA, está presente somente o registro dos fluxos de pacote direcionadas da rede para vítima e da vítima para a rede. Para isso, nos dois cenários selecionados do base de dados da CTU-13 (cenários 10 e 11), extraiu-se com base nos arquivos de descrição, somente o tráfego de dados "da vítima" e "para a vítima" utilizando TCPDUMP. Após este tratamento, selecionaram-se os períodos a serem processados pelo CGP.

Para cada base de dados, selecionou-se um período de dois minutos que compreendem 30 segundos antes do início dos ataques e 1 minuto e meio subsequentes. Os trinta segundos iniciais, que não correspondem ao início do ataque, foram adicionados a fim de avaliar um intervalo de tempo total no qual o ataque ainda não teve início. O intervalo subsequente de um minuto e meio durante o ataque foi selecionado a fim de identificar a *botnet* geradora de DDoS nos primeiros um minuto e meio a contar do início do ataque. Isso por assumirmos que caso a *botnet* geradora do ataque seja identificada dentro de um minuto e meio a partir do início do ataque, as contramedidas para mitigar os efeitos dos ataques podem ser tomadas antes de que os serviços sejam negados de fato. A Figura 4.1 ilustra nos cinco cenários os períodos selecionados para avaliação. Nas bases de dados que apresentam ataques DDoS os traços verticais em azul representam o período total de dois minutos avaliados. O primeiro intervalo de trinta segundos que precede início do ataque está representado pela cor laranja. Os demais um minuto e meio subdivididos em trinta segundos cada correspondem ao período de ataque estão representados pelas cores verde, azul e lilás respectivamente. Na base de dados que não apresenta ataque DDoS, foram selecionados os mesmos períodos, contudo por não haver ataque não houve seleção específica para processamento. Ao final, cada uma das cinco bases de dados apresenta cinco arquivos de registro de fluxo de dados. O primeiro contém o registro no período de dois minutos selecionado. Os outros quatro contém os mesmos registros divididos em trinta segundos cada.

Para cada arquivo de registro gerado foi aplicado o Passo 1 do método CGP. No Passo 1 foi considerado como amostra a quantidade de pacotes de um host a cada 70 milissegundos (ms). O intervalo de tempo considerado para as observações foi de 70ms, considerando um estudo realizado em 2006 por Gibson onde foi calculado que a média de tempo de ida e volta de um pacote (do inglês, *Round Trip Time* - RTT) mundial é 140ms [Gibson, 2006]. Com isso, como avaliamos de forma passiva somente as mensagens recebidas, somente metade desse tempo deve ser necessário como intervalo para avaliação CGP. Isto torna possível considerar que intervalo utilizado nas observações como sendo suficiente para que uma requisição "legítima" completa seja realizada. A título de experimento o valor do intervalo entre amostras pode ser alterado no script em shell (GTS) que gera a série temporal de entrada para o Passo 2. A Figura 4.2 ilustra o processo de extração e seleção das janelas de tempo avaliadas neste estudo.

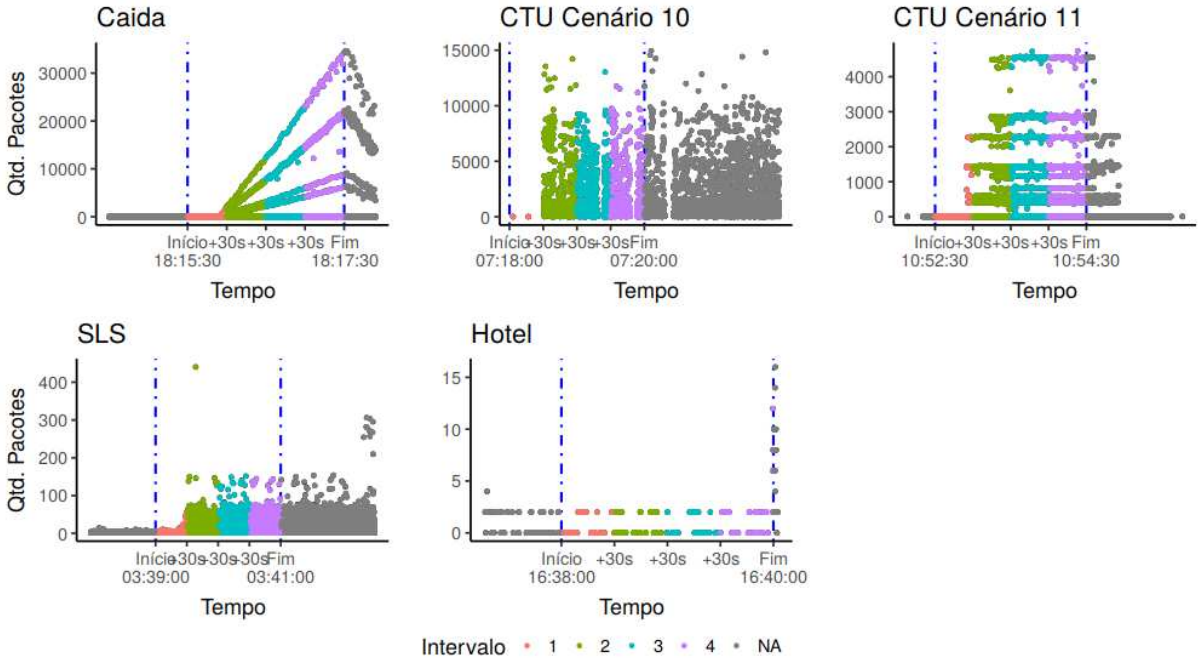


Figura 4.1: Períodos Selecionados

Para o Passo 2, dentro da implementação em Matlab do método CGP, existem dois parâmetros principais a serem configurados. O primeiro é a λ (lambda), que corresponde ao coeficiente de ruído da matriz dada como entrada, e que utilizado para estimar o ruído estatístico gerado pelos cálculos do método CGP. Este valor varia de acordo com a magnitude dos valores das células da série temporal de entrada. Por orientação dos autores do método CGP, utiliza-se valores entre 15 e 21. O outro parâmetro configurado é a quantidade máxima de iterações do método. Esta variável a princípio não deveria existir, a considerar que uma vez que os erros dos modelos estimados pelas autoregressões comecem a convergir as iterações devem parar. Contudo, por se ter acesso a uma versão preliminar da implementação do método CGP, os autores do processo disponibilizaram essa variável para limitar a quantidade de iterações realizadas onde foi utilizado o número máximo de 15 iterações.

A título de comparação, foram realizados testes utilizando as mesmas bases de dados, com os mesmos tratamentos, porém utilizando métodos convencionais de cálculo de correlação entre varáveis. Os métodos utilizados foram o coeficiente de correlação de Pearson, e o coeficiente de correlação de postos de Spearman. Após realizados os cálculos utilizando implementações em R destes conjuntos de dados, os resultados foram comparados com os resultados obtidos com o CGP para que fosse possível diferenciar a acurácia do CGP em relação a estes métodos para estimativa de autocorrelação.

4.2 ANÁLISES

Após o tratamento e organização mencionados na Seção 4.1, todos os arquivos de captura tratados de todos os cenários foram processados pelo método proposto e ilustrado na Figura 3.3. Os resultados de cada cenário estão descritos da seguinte forma. A Subseção 4.2.1 apresenta os resultados obtidos após o processamento de cada um dos dois cenários avaliados correspondentes ao base de dados da CTU-13. A Subseção 4.2.2 apresenta os resultados obtidos após o processamento do base de dados da CAIDA. A Subseção 4.2.3 apresenta os resultados obtidos após o processamento do base de dados SLS. A Subseção 4.2.4 apresenta os resultados

Figura 4.2: Processo de Seleção das Janelas

obtidos após o processamento do base de dados HOTEL. Para todos os cenários foram gerados gráficos que auxiliam na análise preliminar dos resultados, os gráficos foram gerados utilizando a biblioteca *igraph* disponibilizada pelo R. Foram também geradas respectivas matrizes de confusão nas base de dados que apresentam informações precisas à respeito dos *bots* para resumir a qualidade das classificações entre a aplicação do método CGP e os demais métodos de cálculo de coeficiente de correlação mencionados na Seção anterior.

4.2.1 Dataset da CTU

Em ambos os cenários da CTU, de acordo com o arquivo de descrição o Rbot é utilizado para realizar os ataques a principal diferença entre os dois cenários da CTU avaliados é a quantidade de *bots* utilizados, sendo que no cenário 10 são utilizados 10 *bots*, e no cenário 11 são utilizados 3 bots. Mais detalhes estão a seguir.

4.2.1.1 Cenário 10

Neste cenário os dez *hosts* infectados (*bots*) fazem parte da mesma sub-rede. Após a infeção manual desses *hosts* às 12:18:15 CEST (07:18:15 BRST) é dado início um ataque DDoS contra a vítima. Por volta de 12:31:31 CEST (07:31:31 BRST) o ataque foi interrompido manualmente pelos autores do dataset. Os autores mencionam que o ataque foi bem sucedido devido ao fato deles não conseguirem resposta do servidor para serviços como “ssh” ou “ping”. Os testes nesse cenário foram realizados utilizando a janela de tempo das 12:18:00 CEST (07:18:00 BRST) até as 12:21:59 CEST (07:21:59 BRST) como descrito na Seção 4.1.0.2 e ilustrado na Figura 4.1. A Figura 4.3 mostra os resultados para o período de dois minutos totais, considerando a matriz de influências sem o tratamento citado na Seção 3.2.0.2 e com o tratamento.

No período de dois minutos avaliado, dez *hosts* foram detectados, dentre os quais sete apresentaram interrelações acima do limiar. Nos trinta segundos que precedem o ataque ilustrado na Figura 4.1 pela cor laranja, quatro *hosts* apresentaram interrelações acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor verde, sete *hosts* apresentaram interrelações acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor azul, nenhum *host* apresentou

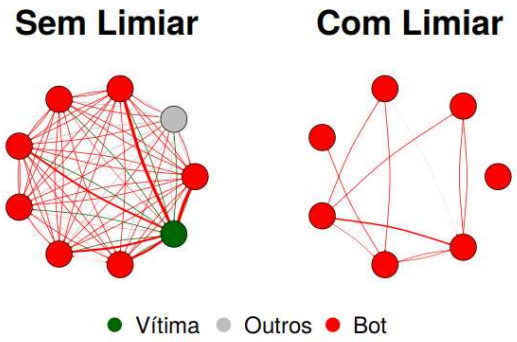


Figura 4.3: Resultado cenário 10

interrelações acima do limiar. Nos últimos trinta segundos subsequentes ilustrados pela cor lilás, sete *hosts* apresentaram interrelações acima do limiar. Após análise aprofundada nos arquivos de captura, constatou-se que dentre os dispositivos que apresentaram interrelações acima do limiar em todos os períodos avaliados, apenas um não estava descrito como sendo *bot*, sendo neste caso a vítima. Os demais *hosts* eram de fato os *bots* ativos nos períodos avaliados. A Tabela 4.6 ilustra na forma de matriz de confusão das classificações nos cinco períodos avaliados e os demais métodos comparados. As matrizes apresentam três linhas e três colunas. O valor da primeira célula corresponde aos períodos avaliados dentro dos cenários. Estes valores correspondem aos intervalos ilustrados na Figura 4.1 de dois minutos delimitados pelis traços verticais em azul, os trintas segundos antes do ataque ilustrados pela cor laranja, os trinta segundos do início dos ataques ilustraods pela cor verde, e os demais trinta segundos subsequentes ilustrados pelas cores azul e lilás respectivamente. Os valores das duas colunas subsequentes correspondem a classificação real dos *hosts* de acordo com a descrição na base de dados. As duas linhas subsequentes correspondem à classificação atribuída aos *hosts* pelo método.

2 min	BOT	Não BOT
BOT	7	0
Não BOT	3	2
30s[1]	BOT	Não BOT
BOT	0	0
Não BOT	0	2
30s[2]	BOT	Não BOT
BOT	7	0
Não BOT	3	2
30s[3]	BOT	Não BOT
BOT	5	0
Não BOT	5	2
30s[4]	BOT	Não BOT
BOT	7	0
Não BOT	3	2

Tabela 4.3: CGP

2 min	BOT	Não BOT
BOT	7	1
Não BOT	3	1
30s[1]	BOT	Não BOT
BOT	0	0
Não BOT	0	2
30s[2]	BOT	Não BOT
BOT	7	1
Não BOT	3	1
30s[3]	BOT	Não BOT
BOT	6	1
Não BOT	4	1
30s[4]	BOT	Não BOT
BOT	6	1
Não BOT	4	1

Tabela 4.4: Pearson

2 min	BOT	Não BOT
BOT	7	1
Não BOT	3	1
30s[1]	BOT	Não BOT
BOT	0	0
Não BOT	0	2
30s[2]	BOT	Não BOT
BOT	7	1
Não BOT	3	1
30s[3]	BOT	Não BOT
BOT	6	1
Não BOT	4	1
30s[4]	BOT	Não BOT
BOT	5	1
Não BOT	5	1

Tabela 4.5: Spearman

Tabela 4.6: Matrizes de Confusão dos Cenário 10

4.2.1.2 Cenário 11

Neste cenário, os três *hosts* infectados (*bots*) fazem parte da mesma sub-rede. Após a infeção manual desses *hosts* às 15:52:39 CEST (11:52:39 BRST) inicia um ataque DDoS contra a vítima com um *bot* e por volta de 15:52:58 CEST (11:52:58 BRST) outro *bot* é adicionado ao ataque. Logo após o segundo *bot* iniciar o ataque, os autores do dataset registraram que o ataque foi bem sucedido. O ataque então foi interrompido em 15:54:44 CEST (11:54:44 BRST). Os testes nesse cenário foram realizados utilizando a janela de tempo das 15:52:30 CEST (11:52:30 BRST) até as 15:54:30 CEST (11:54:30 BRST). A Figura 4.4 mostra os resultados para o período de dois minutos totais, considerando a matriz de influências sem o tratamento citado na Seção 3.2.0.2 e com o tratamento.

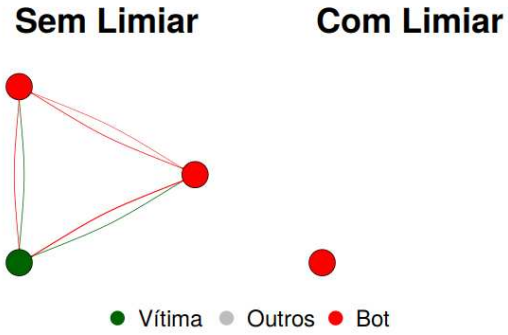


Figura 4.4: Resultado cenário 11

Neste cenário, durante o período de dois minutos avaliados, seis dispositivos foram detectados, dentre os quais três apresentaram interrelações acima do limiar. Nos trinta segundos que precedem o ataque ilustrado na Figura 4.1 pela cor laranja, nenhum *host* apresentou qualquer interrelação acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor verde, dois *hosts* apresentaram interrelações acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor azul, três *hosts* apresentaram interrelações acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor lilás, três *hosts* apresentaram interrelações acima do limiar. Após análise mais aprofundada nos arquivos de captura, constatou-se que somente dois dos três*bots* estava de fato ativo durante o ataque, o outro *bot* não apresentou nenhuma atividade durante a captura. A Tabela 4.6 ilustra na forma de matriz de confusão das classificações nos cinco períodos avaliados e os demais métodos comparados. As matrizes apresentam três linhas e três colunas. O valor da primeira célula corresponde aos períodos avaliados dentro dos cenários. Estes valores correspondem aos intervalos ilustrados na Figura 4.1 de dois minutos delimitados pelis traços verticais em azul, os trintas segundos antes do ataque ilustrados pela cor laranja, os trinta segundos do início dos ataques ilustraods pela cor verde, e os demais trinta segundos subsequentes ilustrados pelas cores azul e lilás respectivamente. Os valores das duas colunas subsequentes correspondem a classificação atribuída aos *hosts* pelo método. As duas linhas subsequentes correspondem a classificação real dos *hosts* de acordo com a descrição na base de dados.

Nos dois cenários a aplicação do método CGP com o limiar dos quartis foi suficiente para identificar a *botnet* nos primeiros um minuto e meio de ataque. No cenário 10 foi possível identificar sete dos dez *bots* da *botnet* e no cenário 11 foi possível identificar dois dos três *bots* presentes no cenário em questão. Para o cenário 10, todos os sete *hosts* classificados como *bot* eram de fato *bots*, os demais que não foram identificados não tiveram atividade registrada nos arquivos de descrição ou nos arquivos de captura. Para o cenário 11, dois *hosts* foram

2 min	BOT	Não BOT
BOT	2	0
Não BOT	1	2
30s[1]	BOT	Não BOT
BOT	1	0
Não BOT	2	2
30s[2]	BOT	Não BOT
BOT	1	0
Não BOT	2	2
30s[3]	BOT	Não BOT
BOT	1	0
Não BOT	2	2
30s[4]	BOT	Não BOT
BOT	2	0
Não BOT	1	2

Tabela 4.7: CGP

2 min	BOT	Não BOT
BOT	2	1
Não BOT	1	1
30s[1]	BOT	Não BOT
BOT	1	1
Não BOT	2	1
30s[2]	BOT	Não BOT
BOT	2	1
Não BOT	1	1
30s[3]	BOT	Não BOT
BOT	2	1
Não BOT	1	1
30s[4]	BOT	Não BOT
BOT	2	1
Não BOT	1	1

Tabela 4.8: Pearson

2 min	BOT	Não BOT
BOT	2	1
Não BOT	1	1
30s[1]	BOT	Não BOT
BOT	1	1
Não BOT	2	1
30s[2]	BOT	Não BOT
BOT	2	1
Não BOT	1	1
30s[3]	BOT	Não BOT
BOT	2	1
Não BOT	1	1
30s[4]	BOT	Não BOT
BOT	2	1
Não BOT	1	1

Tabela 4.9: Spearman

Tabela 4.10: Matrizes de Confusão dos Cenário 11

classificados *bots* foram detectados e identificados corretamente. O terceiro *bot* não apresentou nenhuma atividade de acordo com os arquivos de descrição e nos arquivos de captura.

4.2.2 Dataset da CAIDA

Neste cenário considerando todo o período de dois minutos processados e avaliados, foram detectados 3747 hosts, incluindo a vítima e possíveis *bots*, neste período foram detectadas e identificadas interrelações acima do limiar entre 2960 *hosts*. Nos trinta segundos que precedem o ataque ilustrados na Figura 4.1 pela cor laranja, doze *hosts* apresentaram interrelação com valores acima do limiar. Nos trinta segundos correspondentes ao período após o início dos ataques, ilustrados pela cor verde, foram detectadas e identificadas interrelações com valores acima do limiar correspondentes a 187 *hosts*. Nos trinta segundos subsequentes, ilustrados pela cor azul, foram detectadas e identificadas interrelações acima do limiar entre 973 *hosts*. E nos últimos trinta segundos processados foram detectadas e identificadas interrelações acima do limiar entre 1587 *hosts*. Nos arquivos de descrição do base de dados da CAIDA existem informações concretas de quem é a vítima. Não foi possível ilustrar o resultado deste cenário devido à grande quantidade de *hosts* identificados como *bot*.

4.2.3 Dataset SLS

Neste cenário foram detectados 15402 hosts durante o processo de avaliação. No período que compreende os dois minutos avaliados como um todo, 11762 *hosts* foram detectados e identificados como *bots*. Nos trinta segundos que precedem o ataque ilustrado na Figura 4.1 pela cor laranja, 3271 *hosts* apresentaram interrelações acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor verde, 6255 *hosts* apresentaram interrelações acima do limiar. Nos trinta segundos subsequentes ilustrados pela cor azul, 9139 *host* apresentou interrelações acima do limiar. Nos últimos trinta segundos subsequentes ilustrados pela cor lilás, 11763 *hosts* apresentaram interrelações acima do limiar. Após análise aprofundada nos arquivos de captura, considerando o *host* vítima e as faixas de IP administradas pelo como descrito pelo autor. Todos os *hosts* classificados como *bots* não faziam parte das faixas de IP correspondentes às que eram

administradas pelo autor do dataset, indicando a precisão da classificação do método. Não foi possível ilustrar o resultado deste cenário devido a grande quantidade de *bots* identificados.

4.2.4 Dataset HOTEL

Neste cenário foram detectados 30 *hosts* durante o processo de avaliação. Contudo, nenhuma correlação foi identificada entre estes *hosts* nos períodos avaliados. Este era o resultado esperado uma vez que este base de dados não apresenta ataques DDoS. A Figura 4.5 ilustra o resultado da aplicação do método CGP nesta captura, onde se observa que não há arestas entre os nós detectados, indicando que não há interrelações no comportamento dos *hosts*.

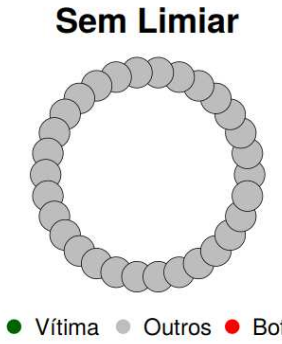


Figura 4.5: Resultado cenário HOTEL

4.2.5 Discussão

Nos cinco cenários avaliados, foi aplicado o método CGP em cinco intervalos de tempo. O primeiro compreende o intervalo de dois minutos correspondentes aos 30 segundos antes do início do ataque e aos um minuto e meio após este período. Os outros quatro intervalos de tempo avaliados correspondem ao mesmo intervalo de tempo de dois minutos divididos por intervalos de 30 em 30 segundos, respectivamente. A Tabela 4.11 apresenta os resultados das análises realizadas nos cinco cenários distintos e nos cinco períodos de tempo mencionados.

Cenário	*hosts* Detectados	2 min	30s[1]	30s[2]	30s[3]	30s[4]	# *bots*	#FP	FN
CTU-13 10	12	7	0	7	5	7	10	0	0
CTU-13 11	5	2	1	1	1	2	3	0	0
CAIDA	3747	2960	12	187	973	1587	NA	NA	NA
SLS	15402	11769	3271	6255	9139	11763	NA	NA	NA
HOTEL	30	0	0	0	0	0	0	0	0

Tabela 4.11: Bots detectados nos períodos de tempo avaliados

Como visto na Tabela 4.11, quando avaliados dos cinco cenários os dois minutos que compreendem dos trinta segundos anteriores ao ataque até um minuto e meio após o início deles, a quantidade de *hosts* classificados como *bots* é inferior à quantidade informada na documentação do ataque. Contudo, quando analisados manualmente os arquivos de captura correspondentes, evidenciou-se que apesar do número inferior de *bots* detectados, estes são de fato os únicos *bots* ativos durante o período avaliado. Isto é, são os únicos *bots* que apresentam atividade na rede durante período de tempo avaliado.

Quando avaliados nos cinco cenários somente os primeiros trinta segundos que correspondem aos trinta segundos após o início do ataque, nos cenários da CTU-13, os *hosts* classificados como *bots* foram detectados em quantidade inferior ao descrito nos arquivos de descrição, isso considerando que somente os cenários CTU-13 apresentam informações sobre a quantidade de *bots* existentes. Nos primeiros trinta segundos correspondentes ao início dos ataques, a quantidade de *bots* detectados foi próxima ao descrito nos cenários selecionados. Neste período, quando feitas análises aprofundadas nos cinco cenários, evidenciou-se que todos os dispositivos classificados como *bots* eram de fato *bots* de acordo com seus respectivos arquivos de descrição. Considerando a taxa de falso positivo obtida nos dois cenários da CTU-13, estima-se que todos os *hosts* classificados como *bots* nas base de dados CAIDA e SLS estão corretos e não apresentam falsos positivos.

Nos trinta segundos subsequentes destes dois cenários, apesar da variação no número *hosts* classificados como *bots*, os *bots* identificados eram de fato os que estavam mais ativos nos períodos observados, o que ratifica os demais resultados. Nos outros dois trinta segundos subsequentes para o base de dados da CAIDA e SLS, o número de *bots* detectados apresentaram um aumento de 748% e 88%, respectivamente. Como não há informações de quem é de fato *bot* neste cenário, por esta informação não estar presente na documentação das bases de dados, assumiu-se que estes são também *bots*.

Para o cenário HOTEL apesar da quantidade de *hosts* detectados na avaliação, nenhuma interrelação que indique a atividade coordenada de *bots* foi identificada. Por isso, nenhum *host* foi classificado como *bot* pelo método, corroborando com o resultado esperado para a aplicação do método CGP neste cenário.

Considerando as análises aprofundadas de cada período avaliado constatou-se que, exceto nos períodos de dois minutos avaliados de todos os cenários, os falsos positivos apresentados nas matrizes de confusão não existem. Isso porque durante os intervalos de trinta segundos avaliados individualmente, os *bots* ignorados não apresentaram atividade na rede local.

Os falsos negativos apontados pelos métodos comparados correspondem aos *hosts* que não são *bots* e que foram classificados como *bots*. Como observado nas matrizes de confusão o CGP não apresentou falso negativo enquanto os demais métodos em alguns momentos chegaram a apresentar. Para o tipo de problema estudado neste trabalho, a ocorrência de falso negativo pode amplificar o efeito do ataque DDoS. Isso porque a principal contramedida em relação aos ataques DDoS é o bloqueio dos *hosts* classificados como *bot*.

Considerando um sistema de defesa em que não se conhece a estrutura da rede, quando observadas as interrelações sem a aplicação do limiar, observou-se que o *host* que apresenta maior influencia de outros hosts é a vítima. Logo, esta informação pode ser utilizada para identificar quem está sendo atacado. Em contrapartida, com o limiar aplicado, apenas os dispositivos que influenciam a vítima ficam expostos, indicando corretamente a *botnet* no período avaliado.

4.3 RESUMO

Este capítulo apresentou a metodologia e ferramentas utilizadas na avaliação do método CGP. Os resultados apontaram em todos os cenários em que se tem a rotulação dos *bots* uma precisão de 100% na detecção dos *bots* em um intervalo de dois minutos considerando 30 segundos antes e um minuto e meio a partir do início efetivo de um ataque DDoS. Nos *datasets* da CTU-13, CAIDA e SLS, identificando o *host* que apresentou sofrer mais influência dos demais *hosts* antes da aplicação do limiar e considerando-o a vítima, observou-se que este *host* não estava presente no resultado após a aplicação do limiar, levando a concluir que os demais *hosts* classificados são de fato *bots*. Na única base de dados em que não há registros de ataques DDoS

o método não identificou nenhuma interrelação entre os *hosts* o que indica que o método é eficaz na detecção e identificação de *bots*. O próximo capítulo apresenta as considerações finais a respeito deste estudo, as dificuldades encontradas, e os trabalhos futuros.

5 CONCLUSÕES

Este estudo teve como finalidade avaliar a adaptação e a eficácia da aplicação do processo causal em grafos (CGP) como método de detecção e identificação de *botnets* geradoras de DDoS. Para isso, foram utilizadas capturas de tráfego de rede contendo ataques DDoS simulados e reais. Mesmo não representando a totalidade de comportamentos e caraterísticas de todas as redes de computadores existentes, as capturas utlizadas neste estudo apresentam cenários de ataques distintos que permitem um melhor diagnóstico na validação da avaliação do método CGP na detecção e identificação de *botnets*.

Considerando a premissa de que um ataque DDoS é iniciado por um atacante que envia através de um *BotMaster* um único comando de ataque e que, o tempo e a geração de dados dos *bots* são fatores constantes, a detecção do ataque e a identificação da *botnet* pode ser exposta a partir da similaridade do padrão de comportamento dos *hosts* durante um ataque. Através da autoregressão e da convolução, o CGP estima a estrutura de interrelações entre os *hosts* de rede durante um ataque. Considerando as premissas mencionadas anteriormente, os *hosts* com as maiores magnitudes de interrelação estão coordenados e, no contexto deste estudo, causam o ataque DDoS.

Foram realizados testes utilizando as mesmas bases de dados, com os mesmos tratamentos, porém com métodos convencionais de cálculo de correlação entre varáveis. O propósito desse comparativo foi verificar se com estes métodos, que são computacionalmente menos custosos, a *botnet* também seria detectada e identificada. Os métodos utilizados foram o coeficiente de correlação de Pearson e o coeficiente de correlação de postos de Spearman. Após a realização dos cálculos considerando os métodos mencionados, constatou-se que os coeficientes de correlação conseguem apresentar as interrelações entre os *hosts*, contudo comparado com o CGP são menos eficazes por apresentarem falsos positivos. Isto é, considerando as mesmas métricas e tratamentos após os cálculos de interrelações, os *hosts* que não faziam parte da *botnet* eram classificados como bots.

Em um primeiro momento, os resultados do CGP não se demonstram tão diferentes dos obtidos com o BotMiner ou o BotGM, por exemplo. Se considerarmos o custo computacional envolvido entre o CGP e os outros dois métodos avaliados pode-se chegar a conclusão que talvez a utilização do CGP não seja viável. Contudo, quando considerados os falsos positivos resultantes dos três métodos avaliados e os problemas que estes falsos positivos podem causar em uma situação real, o CGP se apresenta como uma solução eficaz e viável para a detecção e identificação de *botnets* geradoras de ataques DDoS volumétricos.

Como avaliação complementar à adaptação e a eficácia do uso do método CGP na identificação de *botnets* geradoras de DDoS, foram realizados testes para determinar a quantidade mínima de amostras necessárias para identificação de *botnets*. Após sucessivas reduções no período de tempo avaliado, impactando diretamente na quantidade das amostras nas séries temporais avaliadas, constatou-se que com dez (10) amostras correspondentes a 700ms (milissegundos) de observações, é possível detectar e identificar *hosts* pertencentes à *botnet*. E que quando não aplicado o limiar estabelecido pelo terceiro quartil de todos os valores resultantes da matriz de interrelações resultantes do método CGP, a estrutura de interrelações estimadas indica também quem é a vítima, por se tratar do *host* mais influenciado no período avaliado. Assim, considerando a complexidade computacional do método CGP sendo $O(mn^2)$ onde o m representa as observações das n variáveis observadas na série temporal, o custo do método passa

a ser $O(n^2)$, tornando-o equivalente aos métodos de classificação como o SVM, apresentando uma abordagem diferente na avaliação dos dados.

5.1 TRABALHOS FUTUROS

A realização da avaliação deste estudo abordou a adaptação e eficácia do uso do método CGP para detecção e identifcar *botnets* geradoras de ataques DDoS volumétrico. Contudo, um dos pontos não abordado por falta de bases de dados foi o comportamento do método quando se tem sistemas distribuídos com rotinas automatizadas de sincronização, como *Dropbox*, *Microsoft One Drive*, etc. Este ponto poderia modificar os resultados da avaliação, dependendo da configuração das rotinas de sincronização desses sistemas. Podendo classificar o tráfego desses sistemas como uma *botnet*, o que pode não ser desejado.

Outro ponto importante, é a linguagem de programação em que o método avaliado está implementado. Esta linguagem é de uma aplicação de alto nível e faz com que sua execução possibilite a validação do método e não necessariamente comprove a eficácia do método em relação aos demais. Além disso, se comparado com outros trabalhos, a configuração do computador utilizado na avaliação do método CGP é inferior, podendo influenciar na avaliação da eficácia do método em relação aos demais que foram testados em computadores com mais recursos de processamento e memória.

Considerando os fundamentos da implementação do método CGP utilizada neste estudo, percebe-se a possibilidade ao menos quatro estudos a serem realizados a fim de dar continuidade a este trabalho. A realização de testes com bases de dados que apresentem o rotinas automatizadas de sincronização, a implementação deste método em uma linguagem que permita sua utilização em dispositivos de rede, a otimização do método para aplicação específica em redes de computadores e a implementação do método como módulo de kernel em sistemas UNIX-like, são exemplo desses estudos.

REFERÊNCIAS

[Ahmad et al., 2017] Ahmad, S., Lavin, A., Purdy, S. e Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262(Supplement C):134 – 147. Online Real-Time Learning Strategies for Data Streams.

[Andrychowicz et al., 2016] Andrychowicz, M., Denil, M., Gómez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B. e de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. Em Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I. e Garnett, R., editores, *Advances in Neural Information Processing Systems 29*, páginas 3981–3989. Curran Associates, Inc.

[Angrishi, 2017] Angrishi, K. (2017). Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681*.

[Arun e Selvakumar, 2009] Arun, R. K. P. e Selvakumar, S. (2009). Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment - A Survey on DDoS Attack Tools and Traceback Mechanisms. Em *IEEE International Advance Computing Conference*, páginas 1275–1280.

[Bertino e Islam, 2017] Bertino, E. e Islam, N. (2017). Botnets and internet of things security. *Computer*, 50(2):76–79.

[Bhuyan et al., 2015] Bhuyan, M. H., Bhattacharyya, D. e Kalita, J. (2015). An Empirical Evaluation of Information Metrics for Low-rate and High-rate DDoS Attack Detection. *Pattern Recogn. Lett.*, 51(C):1–7.

[Binkley e Singh, 2006] Binkley, J. R. e Singh, S. (2006). An algorithm for anomaly-based botnet detection. Em *Proceedings of the 2Nd Conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, SRUTI'06, páginas 7–7, Berkeley, CA, USA. USENIX Association.

[Caswell et al., 2003] Caswell, B., Foster, J. C., Russell, R., Beale, J. e Posluns, J. (2003). *Snort 2.0 Intrusion Detection*. Syngress Publishing.

[Chatfield, 2004] Chatfield, C. (2004). *The analysis of time series: an introduction*. CRC Press, Florida, US, 6th edition.

[Choi et al., 2007] Choi, H., Lee, H., Lee, H. e Kim, H. (2007). Botnet Detection by Monitoring Group Activities in DNS Traffic. Em *Proceedings of the 7th IEEE International Conference on Computer and Information Technology*, (CIT), páginas 715–720, Washington, DC, USA. IEEE Computer Society.

[Clarke e Cooke, 1978] Clarke, G. M. e Cooke, D. (1978). *A basic course in statistics*, volume 406. Arnold New York.

[Cohen et al., 2013] Cohen, J., Cohen, P., West, S. e Aiken, L. (2013). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Taylor & Francis.

[Cronin, 1997] Cronin, M. J. (1997). *Global Advantage on the Internet: From Corporate Connectivity to International Competitiveness*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

[Diego et al., 2018] Diego, O. J., Carlos, M. H. e Wilman-Santiago, O. M. (2018). Economic growth and internet access in developing countries: The case of south america. Em *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, página 1–4.

[Farines et al., 2000] Farines, J.-M., da Silva Fraga, J. e de Oliveira, R. S. (2000). *Sistemas de Tempo Real*. Universidade Federal de Santa Catarina.

[Feily et al., 2009] Feily, M., Shahrestani, A. e Ramadass, S. (2009). A survey of botnet and botnet detection. Em *Third International Conference on Emerging Security Information, Systems and Technologies*, páginas 268–273.

[Figueiredo et al., 2007] Figueiredo, M. A. T., Nowak, R. D. e Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597.

[Filho, 2013] Filho, J. (2013). *Análise de Tráfego em Redes TCP/IP: Utilize tcpdump na análise de tráfegos em qualquer sistema operacional*. NOVATEC.

[Fonseca, 2018] Fonseca, A. (2018). Captura de dados realizada em 05/04/2018 das 03:00:00 às 06:00:00 GMT -3 via NetFlows em um cliente da Specialist Linux Solutions, cedido pelo diretor Alexandre Fonseca.

[García, 2011] García, S. (2011). The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic.

[García et al., 2014] García, S., Grill, M., Stiborek, J. e Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45(Supplement C):100 – 123.

[Gibson, 2006] Gibson, B. (2006). TCP limitations on file transfer performance hamper the global Internet. Relatório técnico, Niwot Networks, Inc.

[Goebel e Holz, 2007] Goebel, J. e Holz, T. (2007). Rishi: Identify bot contaminated hosts by irc nickname evaluation. Em *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, páginas 8–8, Berkeley, CA, USA. USENIX Association.

[Gu et al., 2008a] Gu, G., Perdisci, R., Zhang, J. e Lee, W. (2008a). Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. Em *Proceedings of the 17th Conference on Security Symposium*, (SS), páginas 139–154, Berkeley, CA, USA. USENIX Association.

[Gu et al., 2008b] Gu, G., Zhang, J. e Lee, W. (2008b). BotSniffer: Detecting botnet command and control channels in network traffic. Em *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS)*.

[Guerrero et al., 2017] Guerrero, M., Montoya, F. G., Baños, R., Alcayde, A. e Gil, C. (2017). Adaptive community detection in complex networks using genetic algorithms. *Neurocomputing*, 266(Supplement C):101 – 113.

[Hick, 2013] Hick, P. (2013). The CAIDA DDoS Attack 2007 Dataset.

[Kalaivani e Vijaya, 2016] Kalaivani, P. e Vijaya, M. (2016). Mining based detection of botnet traffic in network flow. Em *International Journal of computer Science and information Technology & Security.*, páginas 535–542.

[Karasaridis et al., 2007] Karasaridis, A., Rexroad, B. e Hoeflin, D. (2007). Wide-scale botnet detection and characterization. Em *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, (HotBots), páginas 7–7, Berkeley, CA, USA. USENIX Association.

[Kong et al., 2016] Kong, X., Chen, Y., Tian, H., Wang, T., Cai, Y. e Chen, X. (2016). A novel botnet detection method based on preprocessing data packet by graph structure clustering. Em *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, páginas 42–45.

[Kurose e Ross, 2010] Kurose, J. e Ross, K. (2010). *Redes de computadores e a internet: uma abordagem top-down*. ADDISON WESLEY BRA.

[Lagraa et al., 2017] Lagraa, S., Francois, J., Lahmadi, A., Miner, M., Hammerschmidt, C. e State, R. (2017). BotGM: Unsupervised Graph Mining to Detect Botnets in Traffic Flows. Em *1st Cyber Security in Networking Conference*, páginas 1–8, Rio de Janeiro, Brazil.

[Liu e Kwak, 2010] Liu, J. e Kwak, K. S. (2010). Hybrid security mechanisms for wireless body area networks. Em *2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)*, páginas 98–103.

[Ltkepohl, 2007] Ltkepohl, H. (2007). *New Introduction to Multiple Time Series Analysis*. Springer Publishing Company, Incorporated.

[Lupien et al., 2017] Lupien, N., Grandhi, S. A., Plotnick, L. e Hiltz, S. R. (2017). Wait, did you say no internet?: An exploratory study of the perceived impact of internet outage. Em *ACM Conference on Computer Supported Cooperative Work and Social Computing*, páginas 231–234, New York, NY, USA. ACM.

[Mansfield-Devine, 2016] Mansfield-Devine, S. (2016). Ddos goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare. *Network Security*, 2016(11):7–13.

[Masud et al., 2008] Masud, M. M., Al-khateeb, T., Khan, L., Thuraisingham, B. e Hamlen, K. W. (2008). Flow-based identification of botnet traffic by mining multiple log files. Em *2008 First International Conference on Distributed Framework and Applications*, páginas 200–206.

[Mei e Moura, 2015] Mei, J. e Moura, J. M. F. (2015). Signal processing on graphs: Modeling (causal) relations in big data. *CoRR*, abs/1503.00173.

[Mirkovic et al., 2002] Mirkovic, J., Prier, G. e Reiher, P. (2002). Attacking DDoS at the source. Em *1IEEE International Conference on Network Protocols*, páginas 312–321.

[Montgomery et al., 2006] Montgomery, D. C., Peck, E. A. e Vining, G. G. (2006). *Introduction to Linear Regression Analysis (4th ed.)*. Wiley & Sons.

[Montgomery e Runger, 2003] Montgomery, D. C. e Runger, G. C. (2003). *Applied Statistics and Probability for Engineers*. John Wiley and Sons.

[Perakovic et al., 2015] Perakovic, D., Periša, M. e Cvitić, I. (2015). Analysis of the IoT Impact on Volume of DDoS Attacks.

[Ramachandran et al., 2006] Ramachandran, A., Feamster, N. e Dagon, D. (2006). Revealing Botnet Membership Using DNSBL Counter-intelligence. Em *Proceedings of the 2Nd Conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, (SRUTI), páginas 8–13, Berkeley, CA, USA. USENIX Association.

[Sachdeva et al., 2008] Sachdeva, M., Singh, G., Kumar, K. e Singh, K. (2008). DDoS Incidents and their Impact: A Review.

[Sandryhaila e Moura, 2013] Sandryhaila, A. e Moura, J. M. F. (2013). Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656.

[Schonewille e van Helmond, 2006] Schonewille, A. e van Helmond, D. (2006). The Domain Name Service as an IDS.

[Stein e Weiss, 2016] Stein, E. M. e Weiss, G. (2016). *Introduction to Fourier analysis on Euclidean spaces (PMS-32)*, volume 32. Princeton university press.

[Strayer et al., 2008] Strayer, W. T., Lapsely, D., Walsh, R. e Livadas, C. (2008). *Botnet Detection Based on Network Behavior*, páginas 1–24. Springer US, Boston, MA.

[Thapngam et al., 2011] Thapngam, T., Yu, S., Zhou, W. e Beliakov, G. (2011). Discriminating ddos attack traffic from flash crowd through packet arrival patterns. Em *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, páginas 952–957.

[Zhou e Ye, 2017] Zhou, Y. e Ye, L. (2017). An empirical analysis of the impact of internet finance on china's economic growth: From the perspective of information and communication technology and financial inclusion. Em *International Conference on Service Systems and Service Management*, páginas 1–5.